PyCon US · 24

By the community, for the community

Untold stories from working on Python packaging

Sumana Harihareswara

[This slide deck is annotated with the transcript of Sumana Harihareswara’s PyCon US 2024 closing keynote presentation, delivered May 19th, 2024; for links and references please see http://harihareswara.net/posts/2024/references-pycon-us-keynote/ .]

[CONFERENCE CHAIR MARIATTA'S INTRO]:

So I know Sumana from way back a long time ago, early in my journey in Python, and I [don't know whether you've ever] heard of her name before, but on one of the first day of PyCon US sprints,

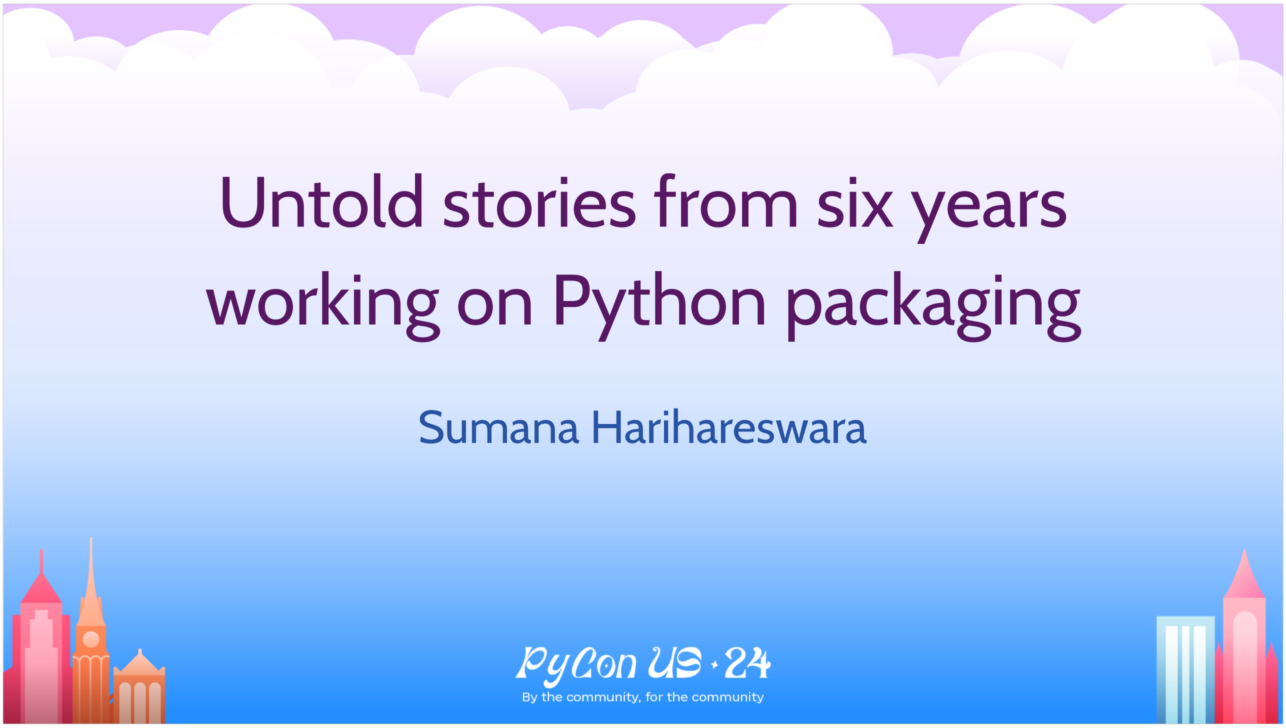
I think a lot of you might have similar feeling that your first time at the conference, you don't know a lot of people, you don't know what, even what is the sprint, you walk to the room and maybe some of you felt lost.

I walked to the sprint room and there was Sumana. She just greeted me and said, "hi, would you like to join our sprint," Zulip’s sprint, if I remember that.

And that just made me realize, you know, that just left a long lasting impression to me, but that's one of the few moments that made me realize, so this is what the Python community is like.

So it is truly an honor for me to be able to invite Sumana Harihareswara here to the keynote stage, so please welcome her.

[APPLAUSE]



Untold stories from six years working on Python packaging

Sumana Harihareswara

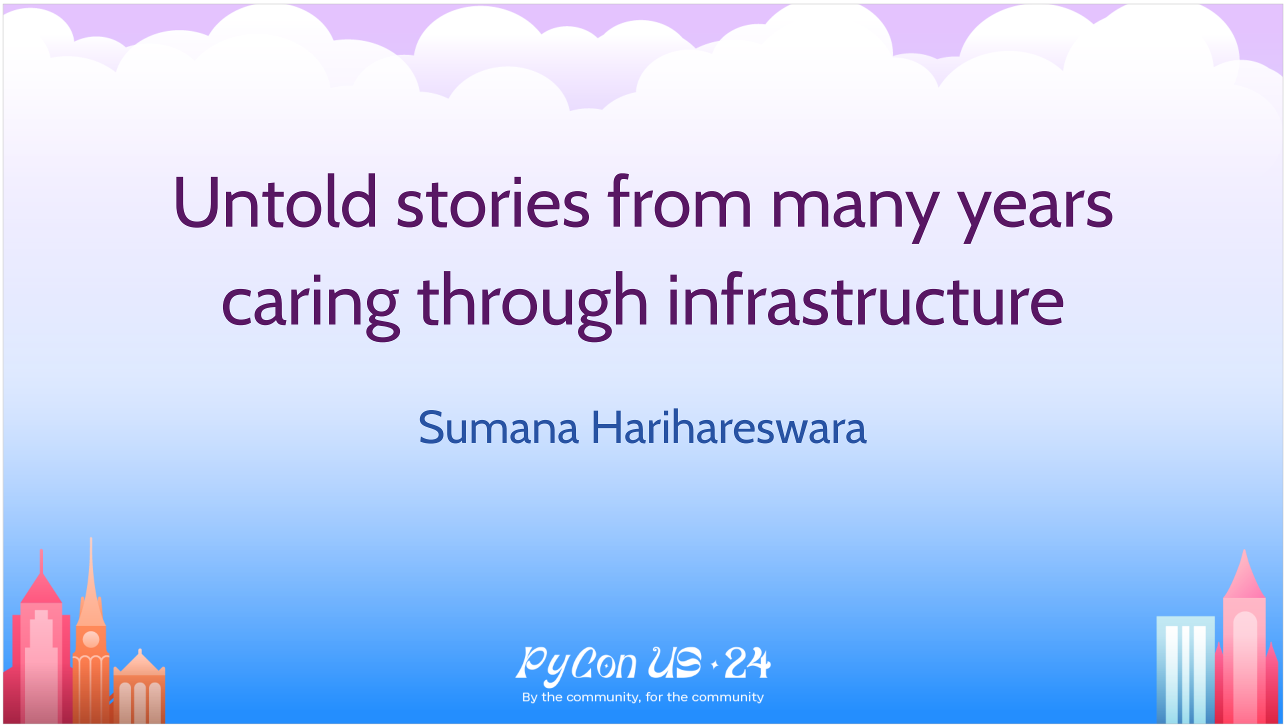
PyCon US · 24

By the community, for the community

Hi everybody.

I used to be the project manager for some things that you use a lot, like pip. When you type 'pip install', I worked on that. Right?

Which means I'm here to tell you untold stories from six years working on Python packaging...



Untold stories from many years caring through infrastructure

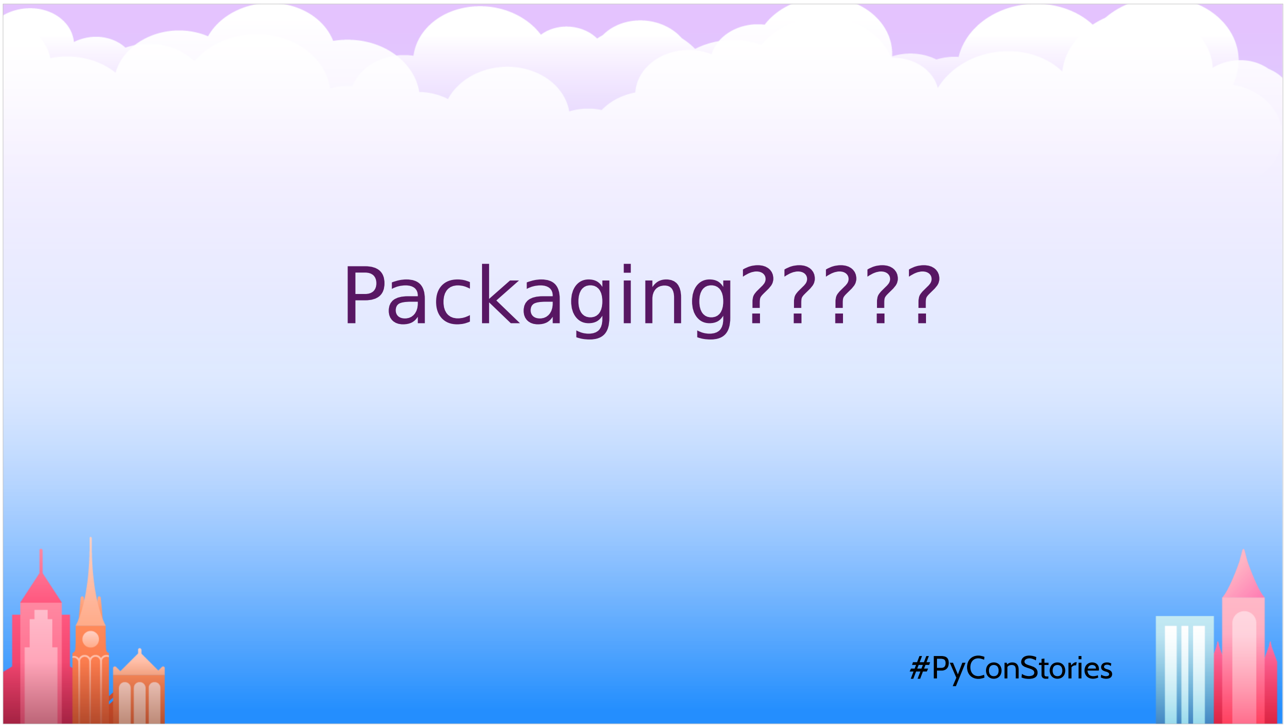
Sumana Harihareswara

PyCon US · 24

By the community, for the community

...which are untold stories from many years caring through infrastructure, because I took care of you. Because infrastructure is a way we take care of each other at scale.

So, for much of the time from about 2017 to 2022, I got paid to work on packaging as a project manager. And you may have a couple of questions.

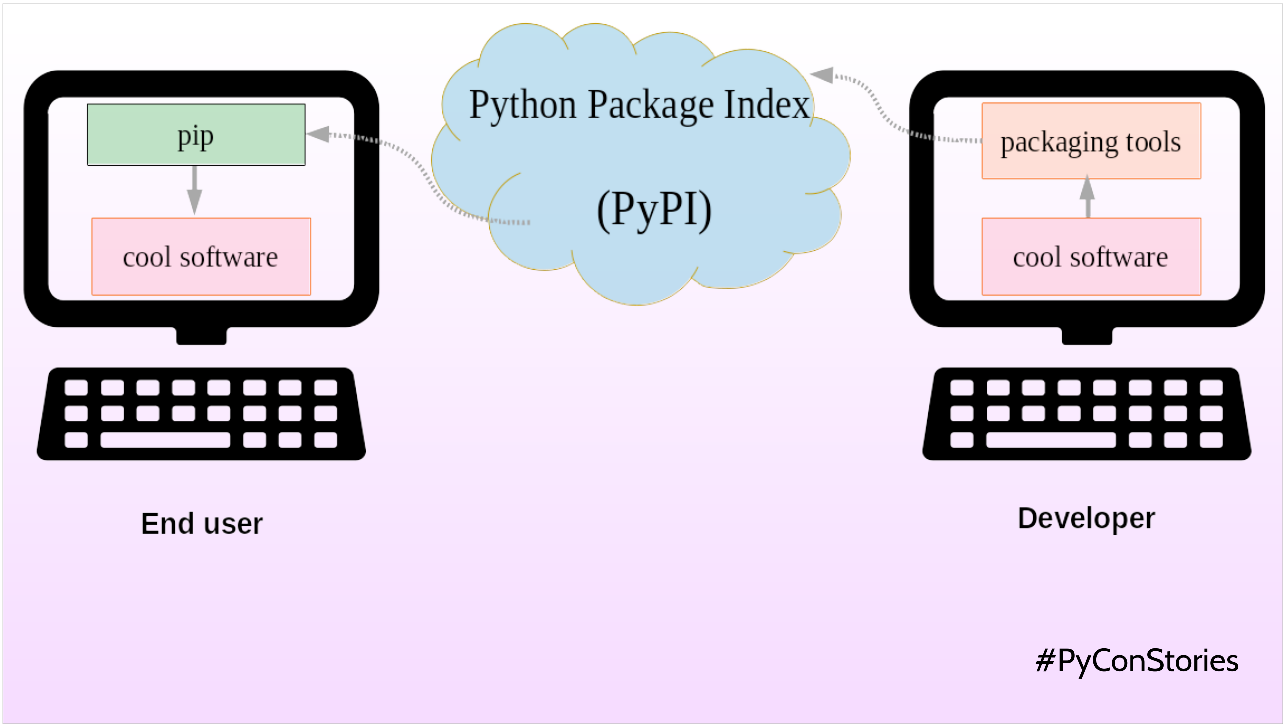


Packaging?????

#PyConStories

Like: what even is packaging????

What do we mean when we talk about Python packaging?



Python Package Index

(PyPI)

pip

cool software

End user

packaging tools

cool software

Developer

#PyConStories

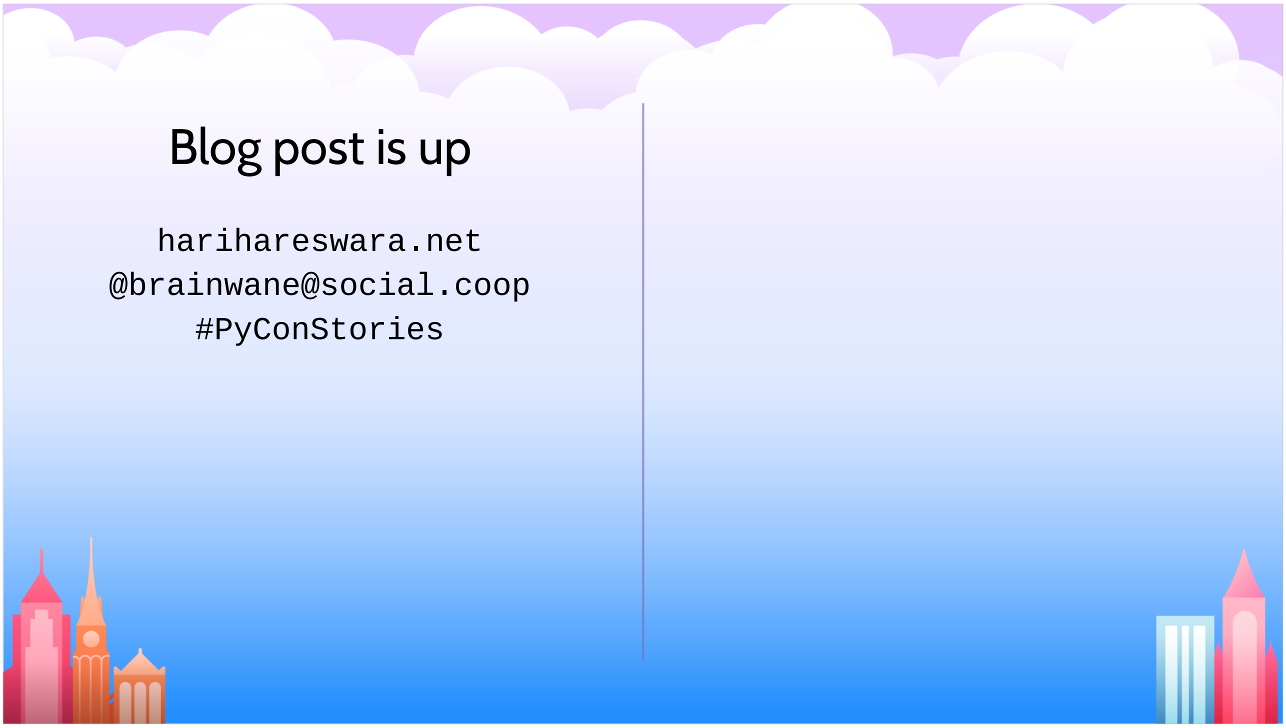
Well, when we say “packaging,” we mean the tools that we use for creating and downloading and installing reusable bits of Python code.

Most of the time when people write software, we don't write everything from scratch, right? We basically borrow and duplicate stuff that other people have done. And this is part of the magic of free and open source software. It's part of how the Internet and most of the technology that you use has been built.

And it's been so useful to have a single website: The Python Package Index, which we call PyPI, is a one-stop shop for all that open source Python code that you might like to make and share, or reuse.

So that's packaging. Developers make cool software and put it in a tidy package and upload it to PyPI, and when you type 'pip install', pip grabs that piece of cool software from PyPI and installs it on your computer.

That is about as far as I'm going to get into the fundamental underpinnings. If you want to know more about the **internals** of how these packaging and distribution tools work on a code level, or the **history** going all the way back to the ‘90s, I'm going to recommend to you several other PyCon talk videos, such as Dustin Ingram's "Inside the Cheeseshop" from 2018.



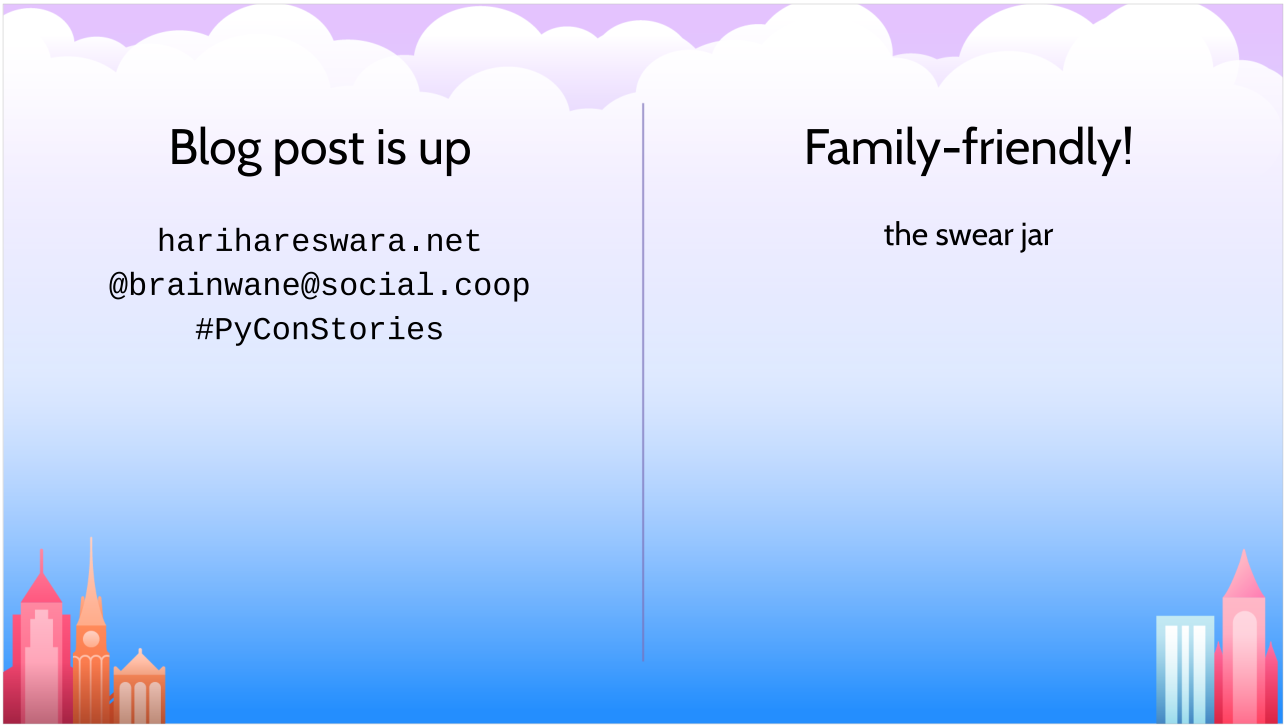
Blog post is up

```
harihareswara.net
@brainwane@social.coop
#PyConStories
```

Oh, this is a good time to share two meta notes about this talk.

One, you don't have to scramble to take notes because I have already posted a blog post with lots of links. You don’t need to – so you can look at the hashtag, #PyConStories. I also put it on #PyConUS.

And here's where to find me online.



Blog post is up

harihareswara.net
@brainwane@social.coop
#PyConStories

Family-friendly!

the swear jar

Second, look. This, like all the talks at PyCon US, is going to be a family-friendly talk, okay?

I'm not going to use swear words. And if I slip up and say a bad word, I am going to put a penny in this swear jar right here. [REMOVES BLACK CLOTH FROM AN OBJECT IN THE CENTER OF THE STAGE, REVEALING THAT IT IS A CLEAR JAR WITH A FEW COINS IN IT; LAUGHTER AND APPLAUSE]

I've explained packaging, but...

Project management??

#PyConStories

... "what even is project management??" is possibly a larger question for many of the, especially the coders in here.

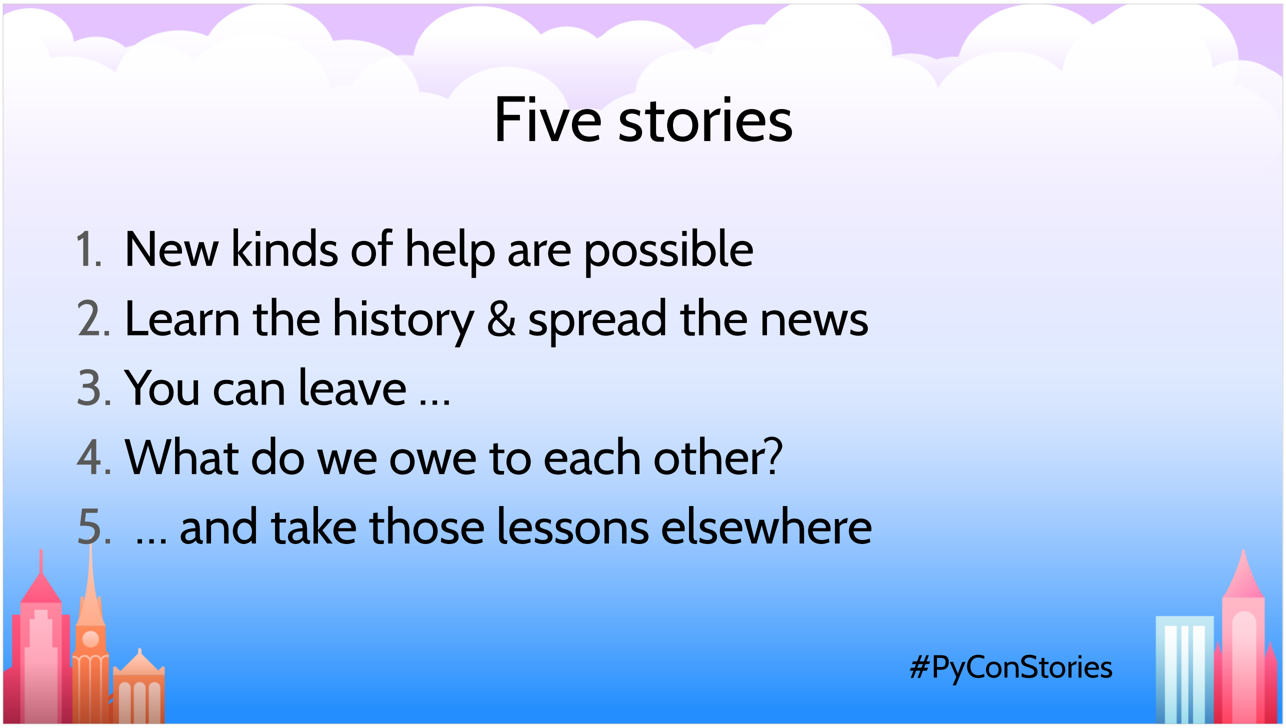
I was project manager for PyPI and later for pip.

One friend, as I was rehearsing this talk, said, "okay, so most people find Python packaging really boring. **And** they think **project management** is also boring. So you're trying to give a keynote about the boring part of a boring thing."

But nothing is boring if you pay the right kind of attention.

And being a project manager means being curious and being stubborn. It means researching what all is happening in a project and getting a sense of the goals, what work is happening towards those goals and what work **isn't** happening, and figuring out the right question to ask and not going away until someone has a good answer.

And it means building trust with a team that we're all actually going to take each other's opinions seriously. And together we're going to make reasonable estimates and roadmaps, and I'll do paperwork and bug triage so you don't have to. I'll communicate on your behalf to funders and users so you can concentrate on what you're doing.

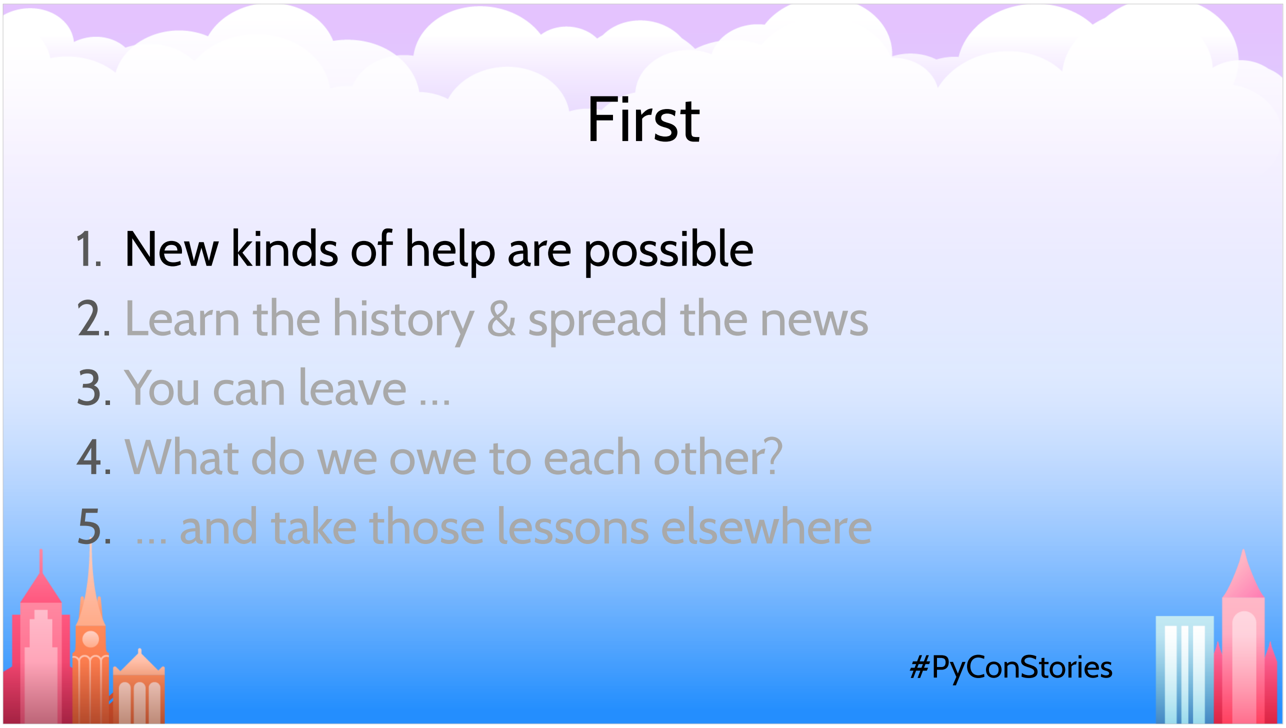


Five stories

1. New kinds of help are possible
2. Learn the history & spread the news
3. You can leave …
4. What do we owe to each other?
5. … and take those lessons elsewhere

#PyConStories

It's been **really** interesting, I think. And I want to share five stories with you about things that happened behind the scenes over the course of those years, as you were using tools that got better.

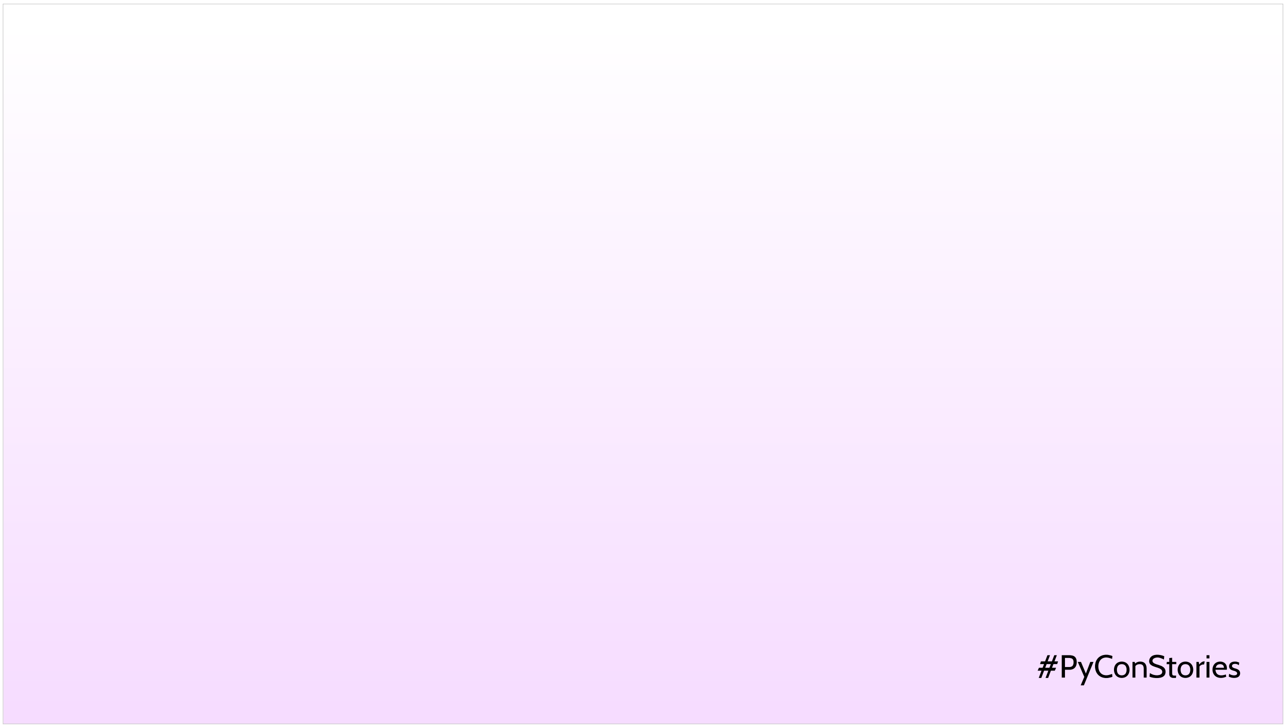


First

1. New kinds of help are possible
2. Learn the history & spread the news
3. You can leave …
4. What do we owe to each other?
5. … and take those lessons elsewhere

#PyConStories

First, just the fact that new kinds of help became possible.



#PyConStories

(This is Python; this is not a blank slide, this is meaningful white space.)
[LAUGHTER]

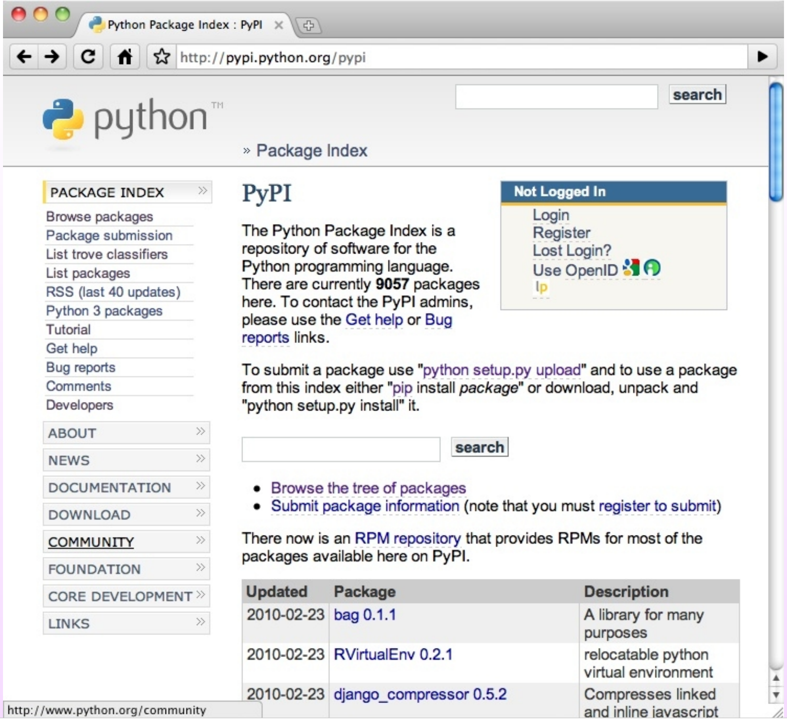
At PyCon US 2016, I ran into an old college friend, Nathaniel Smith, who was working on Python packaging. And I had just started a consultancy doing short-term project management for open source projects. And so I thought, "oh, maybe I could find some clients at PyCon."

And Nathaniel was like, "yeah, yeah, maybe you could work with me on ... NumPy."

Didn't bring up packaging, right? NumPy forms the basis for numerical computing; it's one of the foundational tools of scientific Python.

We were talking about what maybe I could do, working out a proposal, and along the way Nathaniel said, "oh, you know who could use a project manager is PyPI."

Because in 2016...



PyPI

The Python Package Index is a repository of software for the Python programming language. There are currently **9057** packages here. To contact the PyPI admins, please use the Get help or Bug reports links.

To submit a package use "python setup.py upload" and to use a package from this index either "pip install *package*" or download, unpack and "python setup.py install" it.

- Browse the tree of packages
- Submit package information (note that you must register to submit)

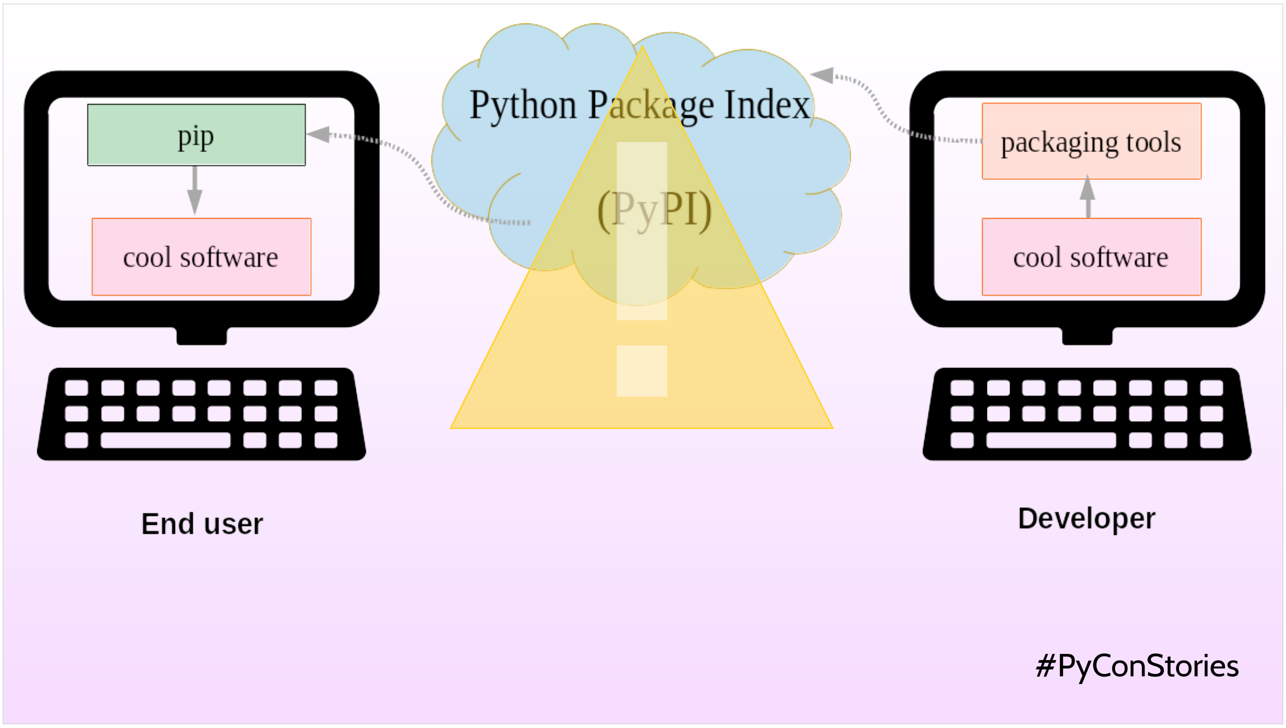
There now is an RPM repository that provides RPMs for most of the packages available here on PyPI.

Updated	Package	Description
2010-02-23	bag 0.1.1	A library for many purposes
2010-02-23	RVirtualEnv 0.2.1	relocatable python virtual environment
2010-02-23	django_compressor 0.5.2	Compresses linked and inline javascript

#PyConStories

… PyPI looked like this.

Maintainer Donald Stufft has said, this "was a weekend hack for a proof of concept that was intended to get quickly replaced by the **real** code. Over a decade later? Yeah, that became critical infrastructure for most Python developers. This code predates modern frameworks by years and it either predates or was a contemporary of WSGI."

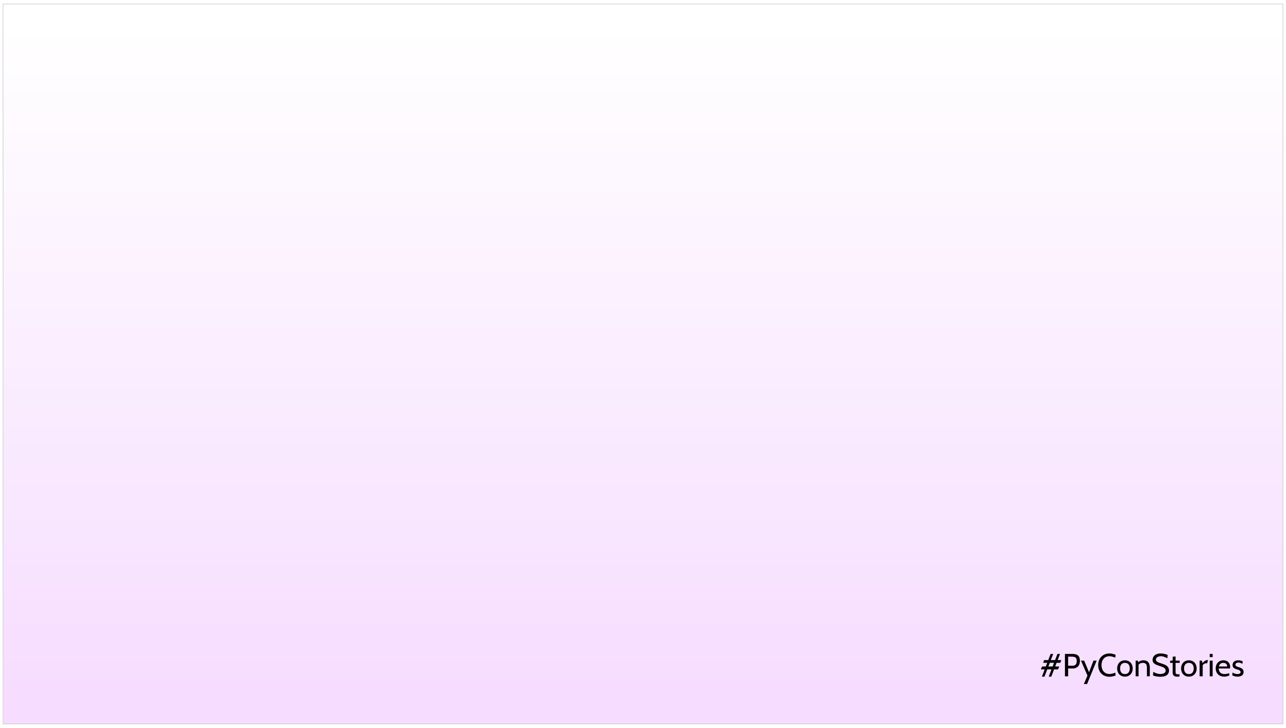


And that legacy site was just terrible for developer sustainability and operation sustainability.

I said I wasn't going to swear. [SIGHS, OPENS SWEAR JAR, PUTS PENNY IN SWEAR JAR; LAUGHTER] (All right. I'll clean up my act, sorry.)

It had tons of technical debt. It had very little automated testing or other tools that would make infrastructure work more doable. So it was unreliable. We couldn't get new people to start contributing to it and we couldn't add new features we needed. **Critical security stuff** like two-factor authentication completely off the table with this particular code base. (And you've probably been hearing about a lot of the work that we've **been** doing [REFERRING TO EARLIER IN THE CONFERENCE], that we had to **build** on a new foundation about, right?)

Donald and others had put in effort for years to write a replacement code base called Warehouse. And there had been a **little** bit of support from some companies, but most of this was by volunteers. They'd made some progress, but we needed to get some concrete, **concentrated** time to get all the way there, so we could redirect all the traffic to the new site and take down the old one. We would need back-end development, DevOps, load testing, front-end development, user experience, research design, tech writing, stakeholder outreach, release management and project management .... and it **really** needed doing.



#PyConStories

But not for me in particular.

In open source, there's often this advice that: if you want to find a project, you should scratch your own itch.

I had no itch here that needed scratching. I was not using these tools.

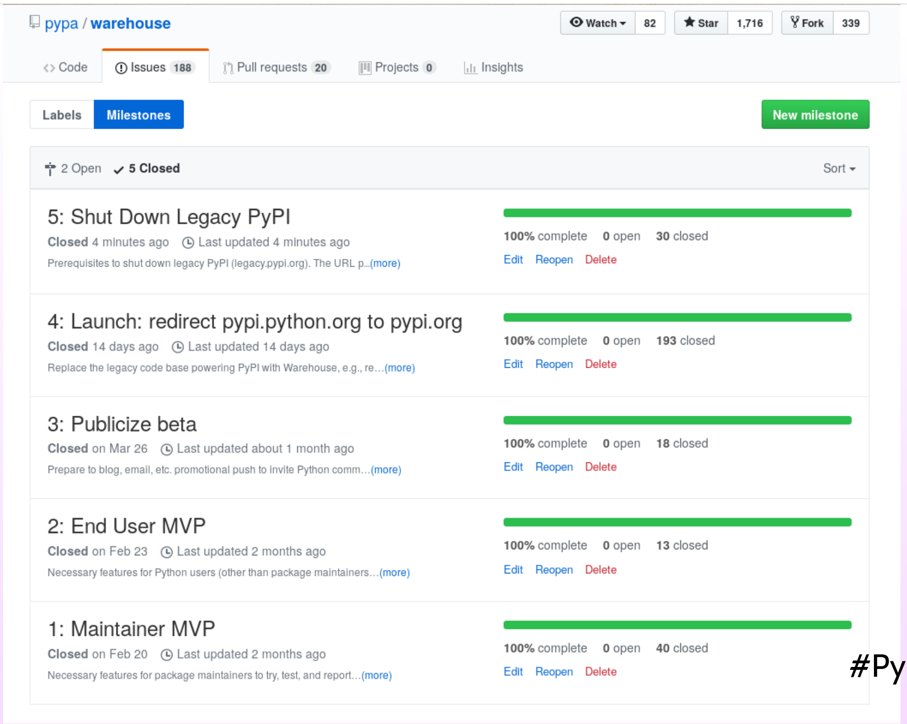
#PyConStories

I – This is the packaging mascot, which is a platypus that's **extremely** enthusiastic about packaging, and this was not me. [LAUGHTER] I was like maybe the ball of string or something.

I was not particularly **passionate** about making Python packages more uploadable, downloadable and installable.

I **did** want to work on something that mattered.

That's what I cared about. That was enough.



pypa / **warehouse**

Watch ▾ 82 | ★ Star 1,716 | Fork 339

<> Code | Issues 188 | Pull requests 20 | Projects 0 | Insights

Labels | **Milestones** | New milestone

2 Open ✓ **5 Closed** Sort ▾

5: Shut Down Legacy PyPI
Closed 4 minutes ago Last updated 4 minutes ago
Prerequisites to shut down legacy PyPI (legacy.pypi.org). The URL p...(more)
100% complete **0** open **30** closed
Edit Reopen Delete

4: Launch: redirect pypi.python.org to pypi.org
Closed 14 days ago Last updated 14 days ago
Replace the legacy code base powering PyPI with Warehouse, e.g., re…(more)
100% complete **0** open **193** closed
Edit Reopen Delete

3: Publicize beta
Closed on Mar 26 Last updated about 1 month ago
Prepare to blog, email, etc. promotional push to invite Python comm…(more)
100% complete **0** open **18** closed
Edit Reopen Delete

2: End User MVP
Closed on Feb 23 Last updated 2 months ago
Necessary features for Python users (other than package maintainers…(more)
100% complete **0** open **13** closed
Edit Reopen Delete

1: Maintainer MVP
Closed on Feb 20 Last updated 2 months ago
Necessary features for package maintainers to try, test, and report…(more)
100% complete **0** open **40** closed
Edit Reopen Delete

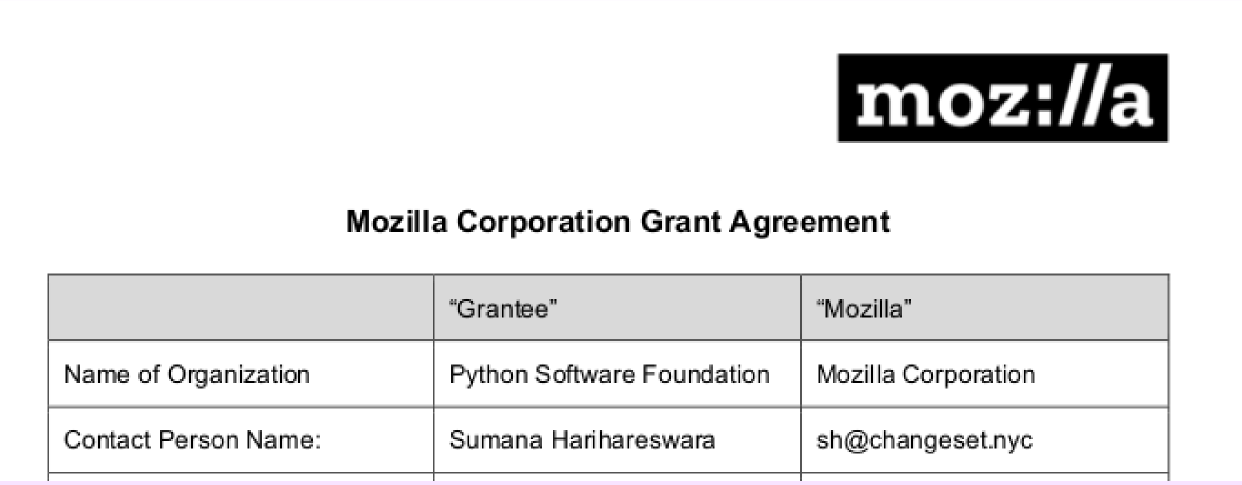
#PyConStories

So I started poking at it.

I did project management.

I – As a volunteer, I cleared out the obsolete parts of the bug tracker.

We made a roadmap for decommissioning the legacy site and launching the new one.



moz://a

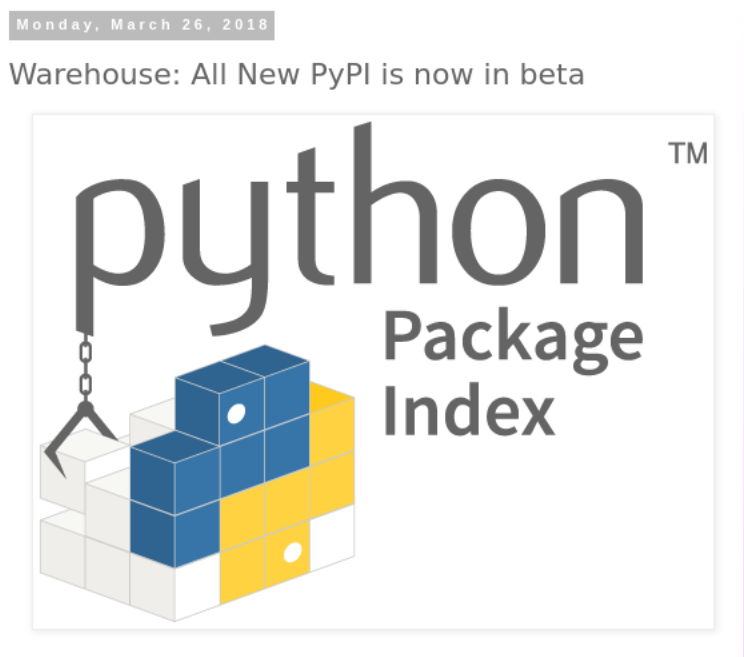
Mozilla Corporation Grant Agreement

	“Grantee”	“Mozilla”
Name of Organization	Python Software Foundation	Mozilla Corporation
Contact Person Name:	Sumana Harihareswara	sh@changeset.nyc

#PyConStories

We figured out a budget.

Submitted a grant proposal and Mozilla Open Source Support, which is on hiatus or possibly defunct, said yes.

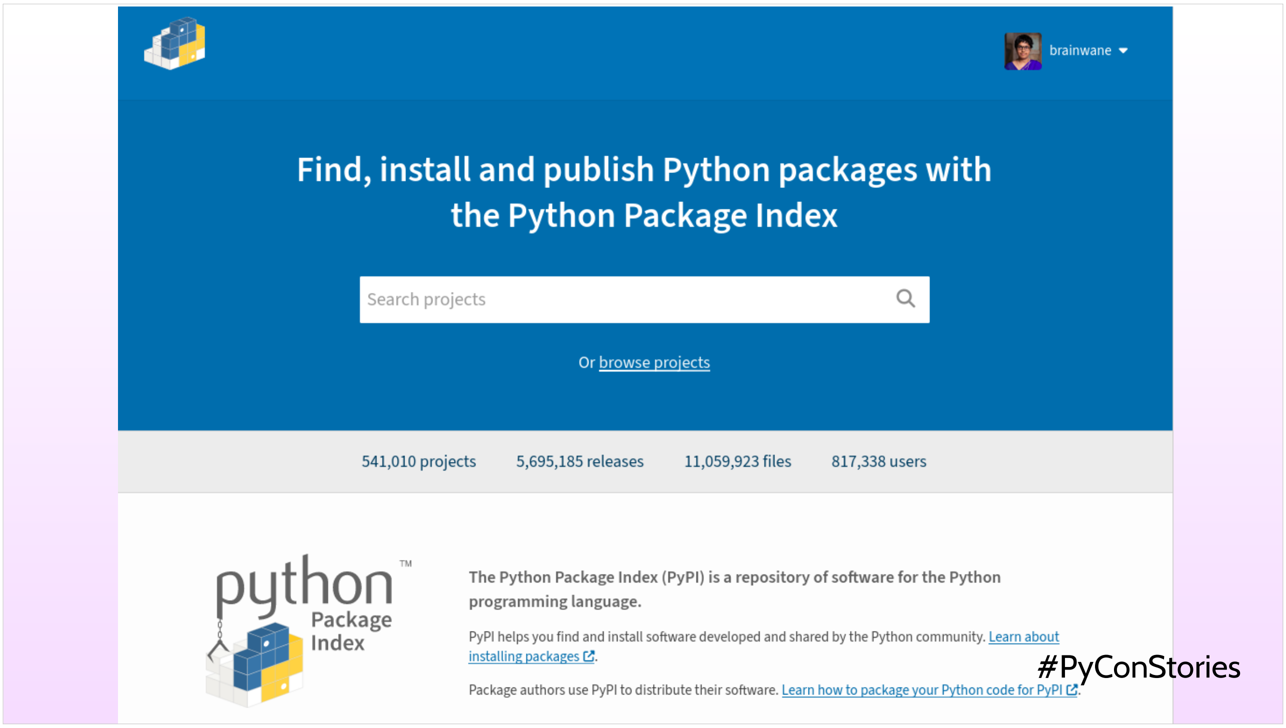


Monday, March 26, 2018

Warehouse: All New PyPI is now in beta

#PyConStories

We used that money, overhauled PyPI. We came in on time and on budget, April 2018, so now ...



Find, install and publish Python packages with the Python Package Index

Or browse projects

541,010 projects 5,695,185 releases 11,059,923 files 817,338 users

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. Learn about installing packages.

Package authors use PyPI to distribute their software. Learn how to package your Python code for PyPI.

#PyConStories

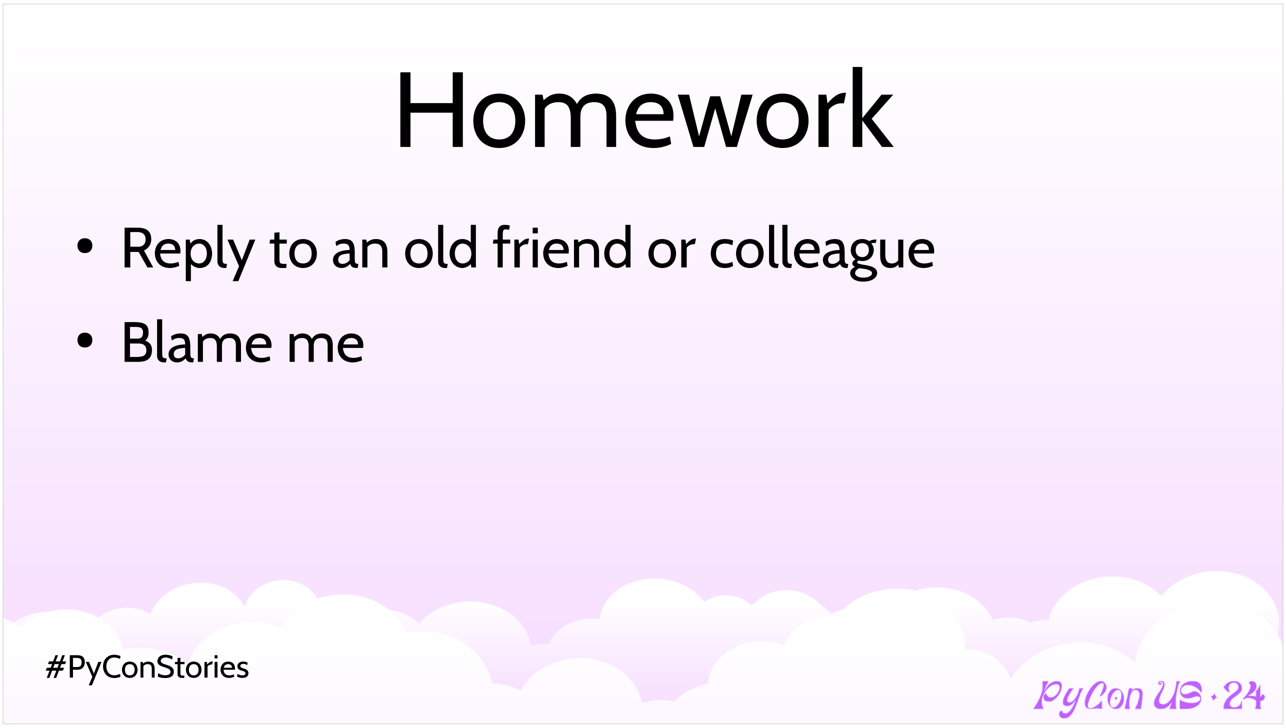
...PyPI looks like this. [APPLAUSE]

And a key reason why all of that happened is that someone I kind of knew saw that PyPI was stuck, and conceptualized it as, "oh, maybe they need a project manager."

When I talked about the idea that new kinds of help are possible, this is what I'm talking about. Knowing that a kind of skill can solve a problem that sometimes people run into makes it possible to realize that a situation isn't just “the way it is.” It's a problem that could be solved. Or at least addressed, or mitigated.

And this goes for project management. This goes for coaching. This goes for researching and writing grant proposals. This goes for user experience research. If you realize that certain skills exist in the world and are useful, then you can think about matching them to situations that could be addressed. You can think about learning those skills, recruiting them, hiring them, finding volunteers, and whether a colleague of yours, maybe, could use that kind of help.

And this is also a story about what gets called the strength of weak ties. You know, Nathaniel and I had not really been in touch that much, right? And that's what made this possible. The people in my circles already **knew** what I was looking for. The people in his circles already knew... But the cross-pollination was possible because of the kind of conversation that emerges at places like PyCon US.



Homework

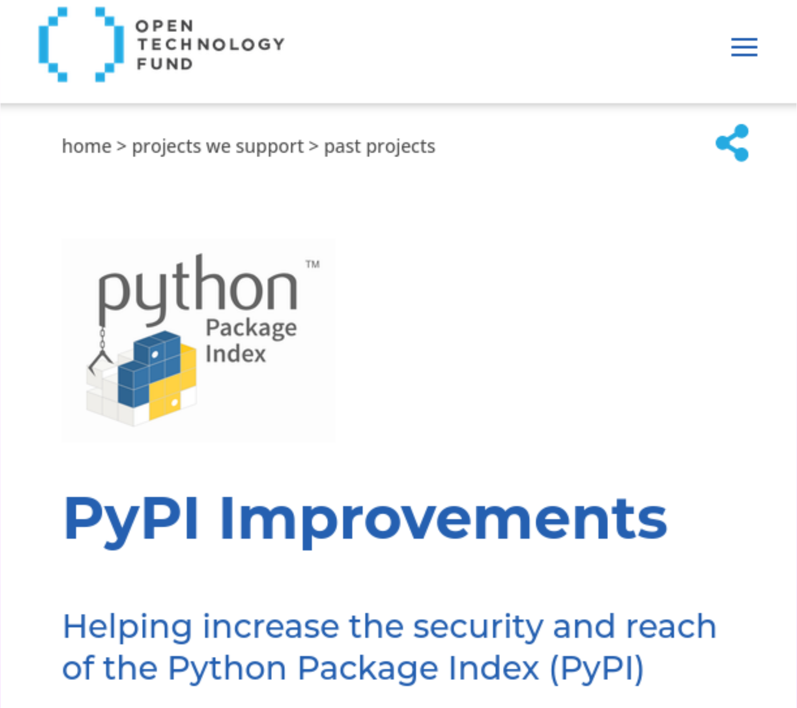
- Reply to an old friend or colleague
- Blame me

#PyConStories

PyCon US · 24

And that's why your first piece of homework from today's keynote: if you have a lingering email in your inbox from an old friend or colleague that you've been **meaning** to reply to, but it feels like it's been **years** – go ahead, hit reply, say "hi."

Blame me. Say there was this jerk of a keynote speaker who said that you should really do this, but say hi.



PyPI Improvements

Helping increase the security and reach of the Python Package Index (PyPI)

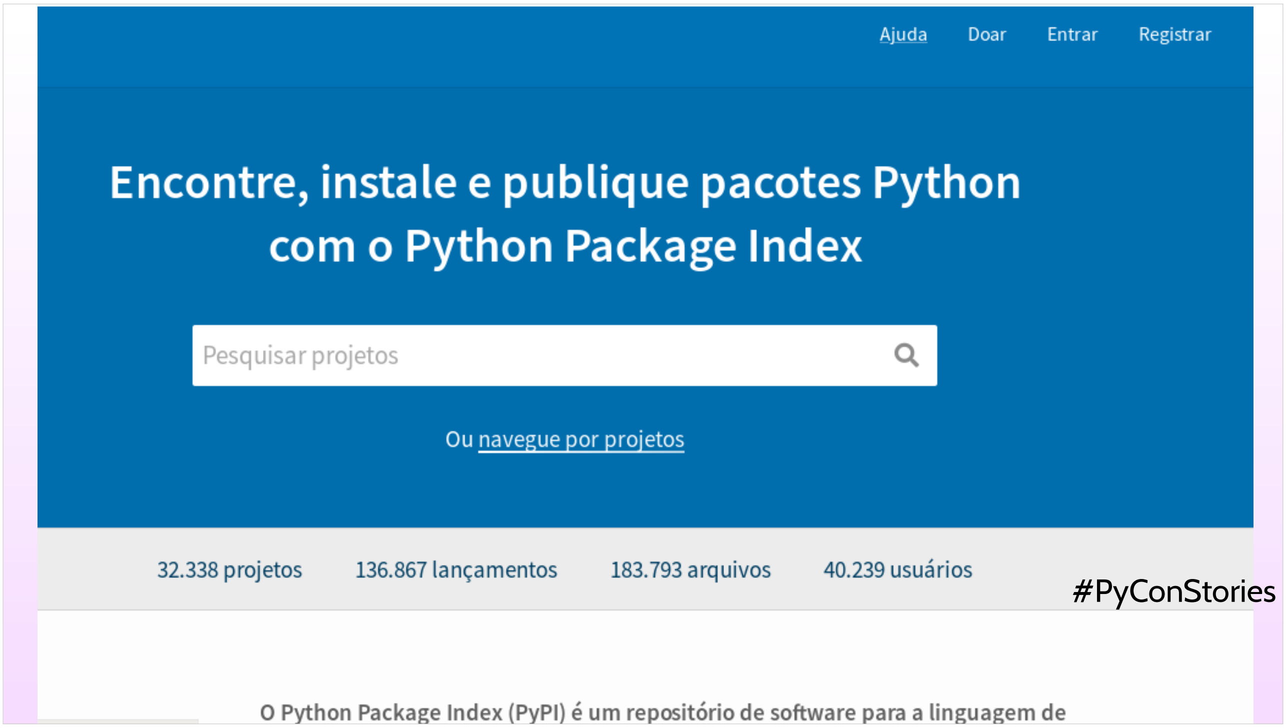
#PyConStories

Oh, we kept on going, right?

After we had overhauled the code base, so now it was Warehouse running, you know, we could actually make more improvements now, right? Because we didn't have all that technical debt.

I found more grant opportunities.

We applied, we got funding to make more improvements. The Open Tech Fund said yes, we could do….



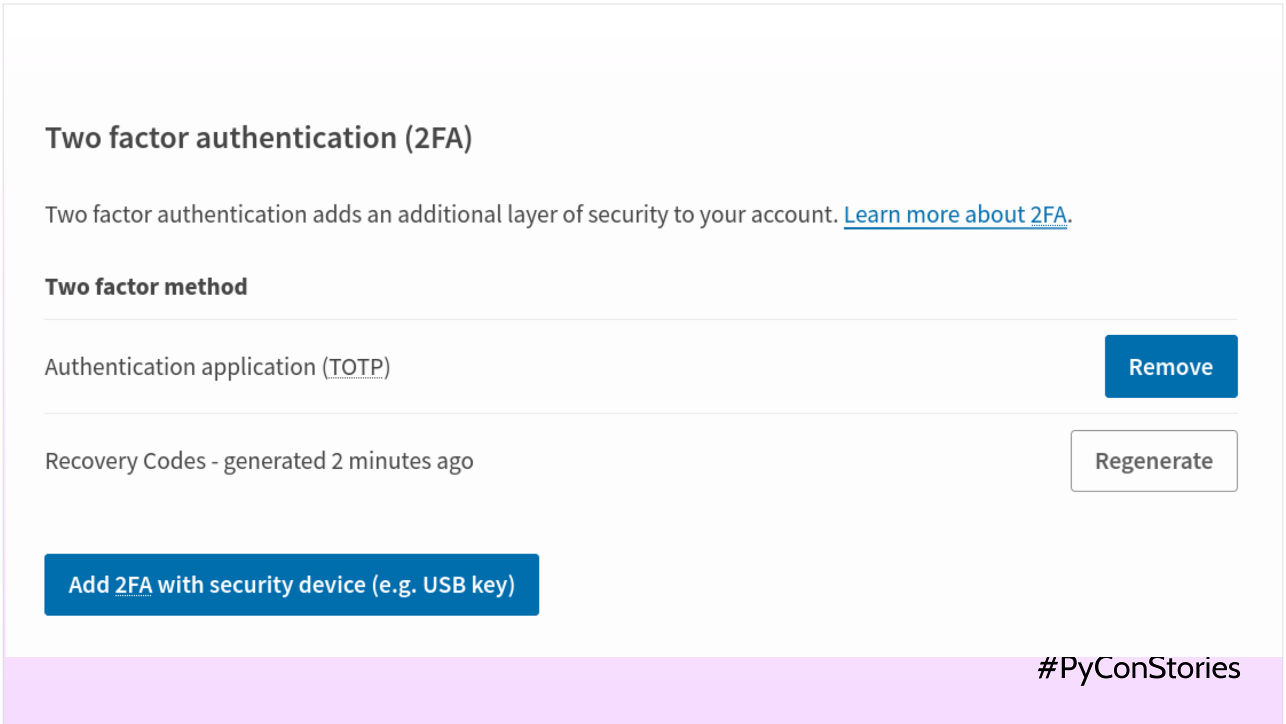
Encontre, instale e publique pacotes Python com o Python Package Index

O Python Package Index (PyPI) é um repositório de software para a linguagem de

#PyConStories

...localization.

PyPI is available in Portuguese.



Two factor authentication (2FA)

Two factor authentication adds an additional layer of security to your account. Learn more about 2FA.

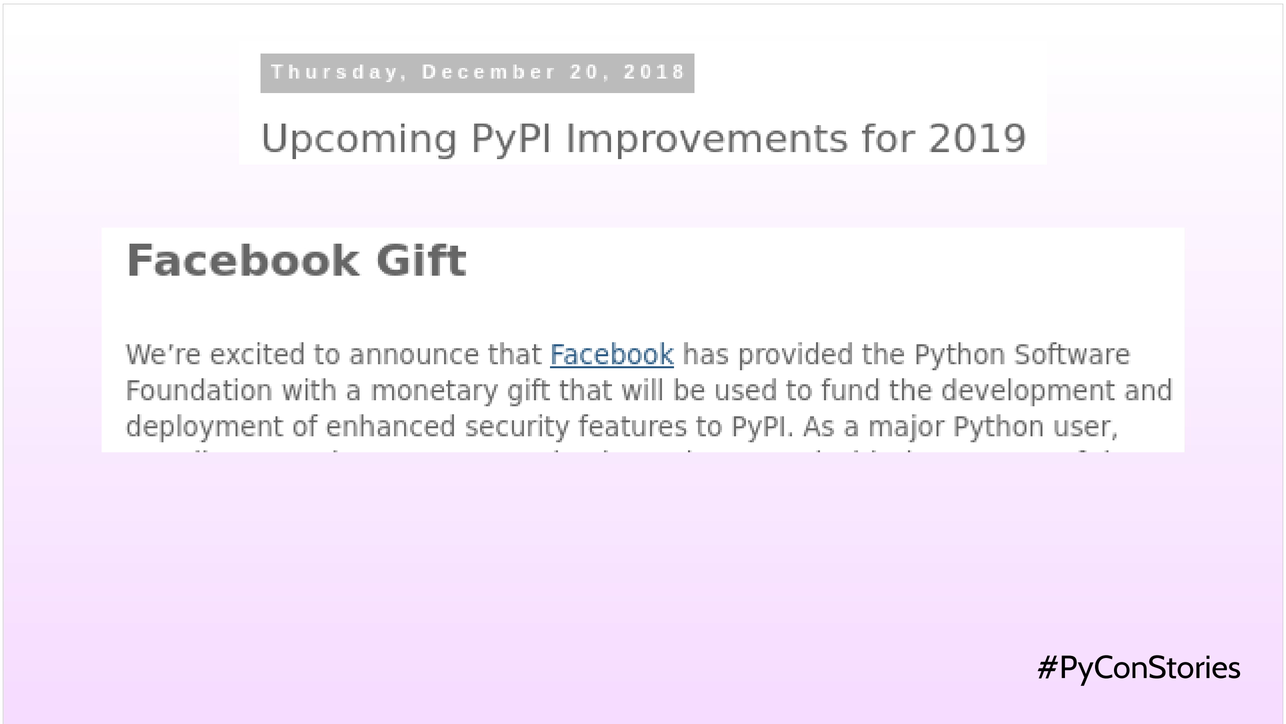
Two factor method

Authentication application (TOTP)

Recovery Codes - generated 2 minutes ago

#PyConStories

We could **finally** add two-factor authentication and a bunch of other security improvements.



Thursday, December 20, 2018

Upcoming PyPI Improvements for 2019

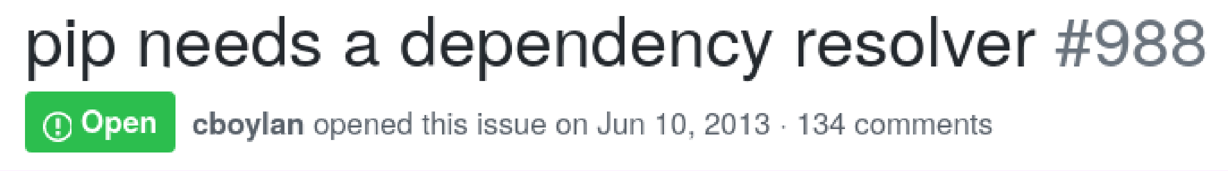
Facebook Gift

We're excited to announce that Facebook has provided the Python Software Foundation with a monetary gift that will be used to fund the development and deployment of enhanced security features to PyPI. As a major Python user,

#PyConStories

Facebook gave us money to do more security work.

And then in 2019, now that I knew more about the packaging ecosystem, I started thinking about...



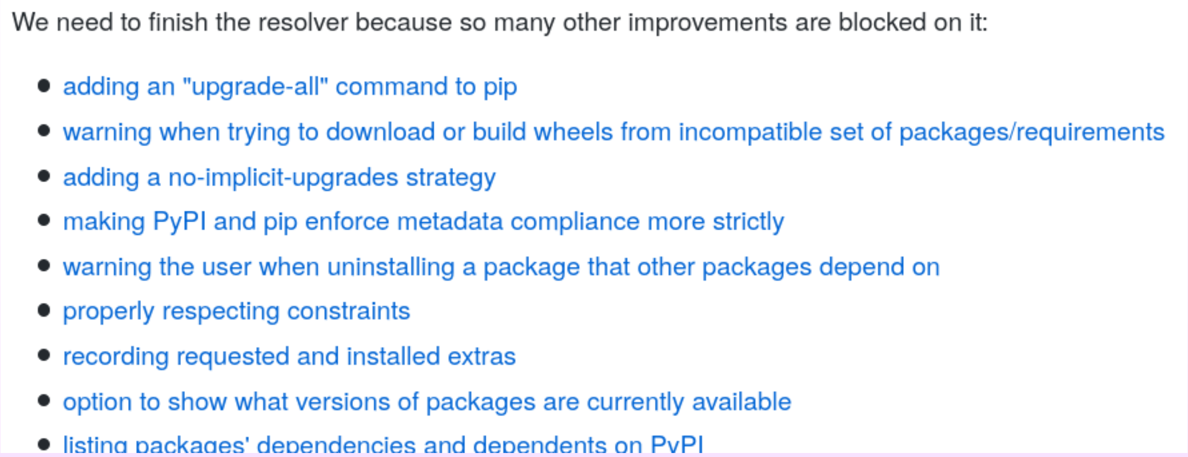
pip needs a dependency resolver #988

Open cboylan opened this issue on Jun 10, 2013 · 134 comments

#PyConStories

... this bug from 2013 in pip.

The heart of pip is, "oh, you want Package X? That means you're going to need Y first." And our old resolver was really unpredictable.



We need to finish the resolver because so many other improvements are blocked on it:

- adding an "upgrade-all" command to pip
- warning when trying to download or build wheels from incompatible set of packages/requirements
- adding a no-implicit-upgrades strategy
- making PyPI and pip enforce metadata compliance more strictly
- warning the user when uninstalling a package that other packages depend on
- properly respecting constraints
- recording requested and installed extras
- option to show what versions of packages are currently available
- listing packages' dependencies and dependents on PyPI

#PyConStories

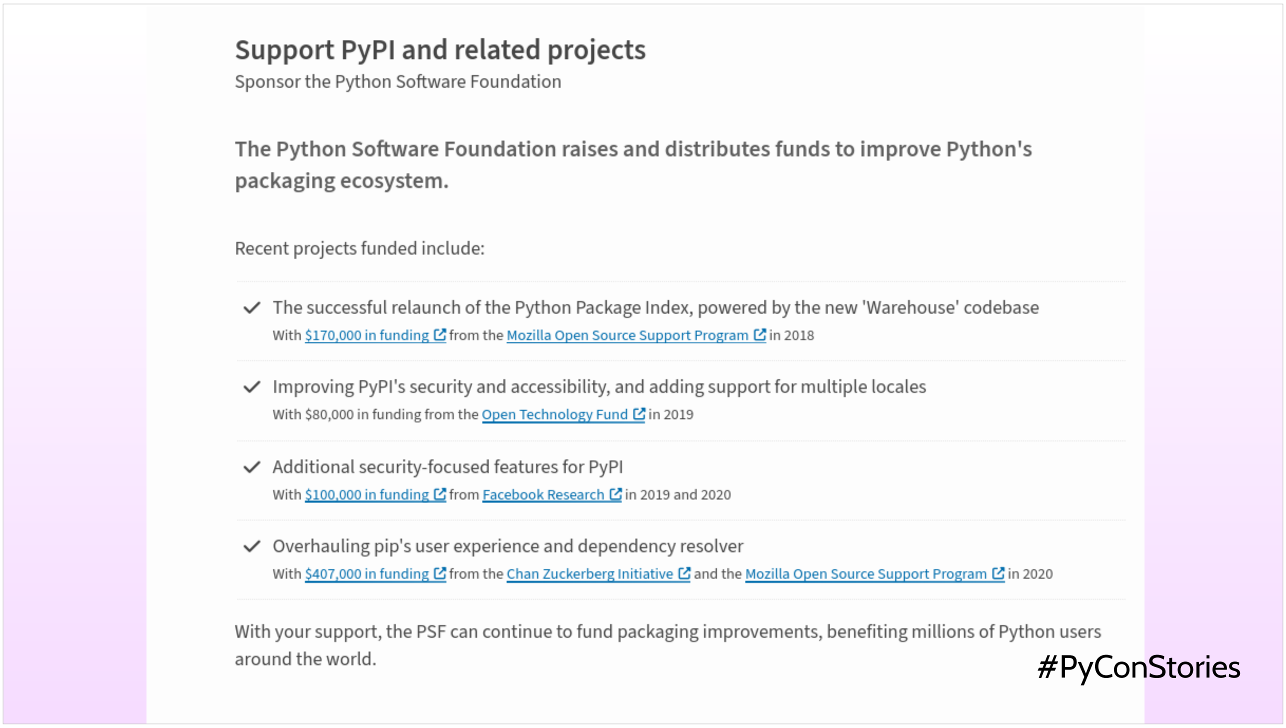
We needed to finish an overhaul of it because a lot of other Python projects were blocked on it, and other features and bug fixes were blocked on it.

Do you build packages?
We want to hear about your wants, needs, a
pip!
If you have another session you want to go t
us to schedule a follow-up call/interview!
CZI
Essential Open Source So
26

So I was the lead author on two grant proposals, one to Mozilla and one to the Chan Zuckerberg initiative.

They both said yes, and so in 2020, I again got paid to do some work on packaging. I was the project manager on a multi-person project to overhaul the resolver at the heart of pip.

We made that happen even though it was 2020. (Oh, boy.)



Support PyPI and related projects

Sponsor the Python Software Foundation

The Python Software Foundation raises and distributes funds to improve Python's packaging ecosystem.

Recent projects funded include:

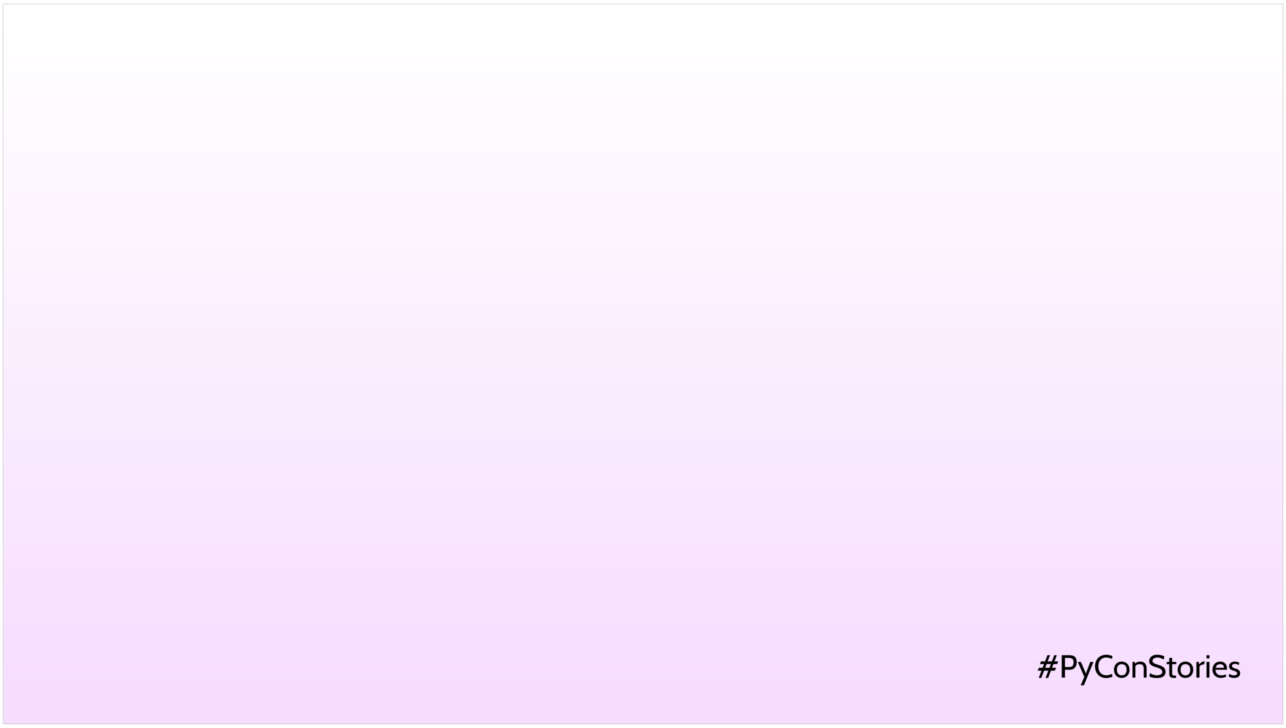
- ✓ The successful relaunch of the Python Package Index, powered by the new 'Warehouse' codebase
  With $170,000 in funding from the Mozilla Open Source Support Program in 2018
- ✓ Improving PyPI's security and accessibility, and adding support for multiple locales
  With $80,000 in funding from the Open Technology Fund in 2019
- ✓ Additional security-focused features for PyPI
  With $100,000 in funding from Facebook Research in 2019 and 2020
- ✓ Overhauling pip's user experience and dependency resolver
  With $407,000 in funding from the Chan Zuckerberg Initiative and the Mozilla Open Source Support Program in 2020

With your support, the PSF can continue to fund packaging improvements, benefiting millions of Python users around the world.

#PyConStories

And so if you go to pypi.org under "sponsor," you'll see this list, which – you know, you don't have to read this right now, right?

It just shows -- the fact that there's so much here that it's hard to show on one slide demonstrates: **When the Python Software Foundation gets funding, it gets results.** [LOTS OF CHEERS AND APPLAUSE]



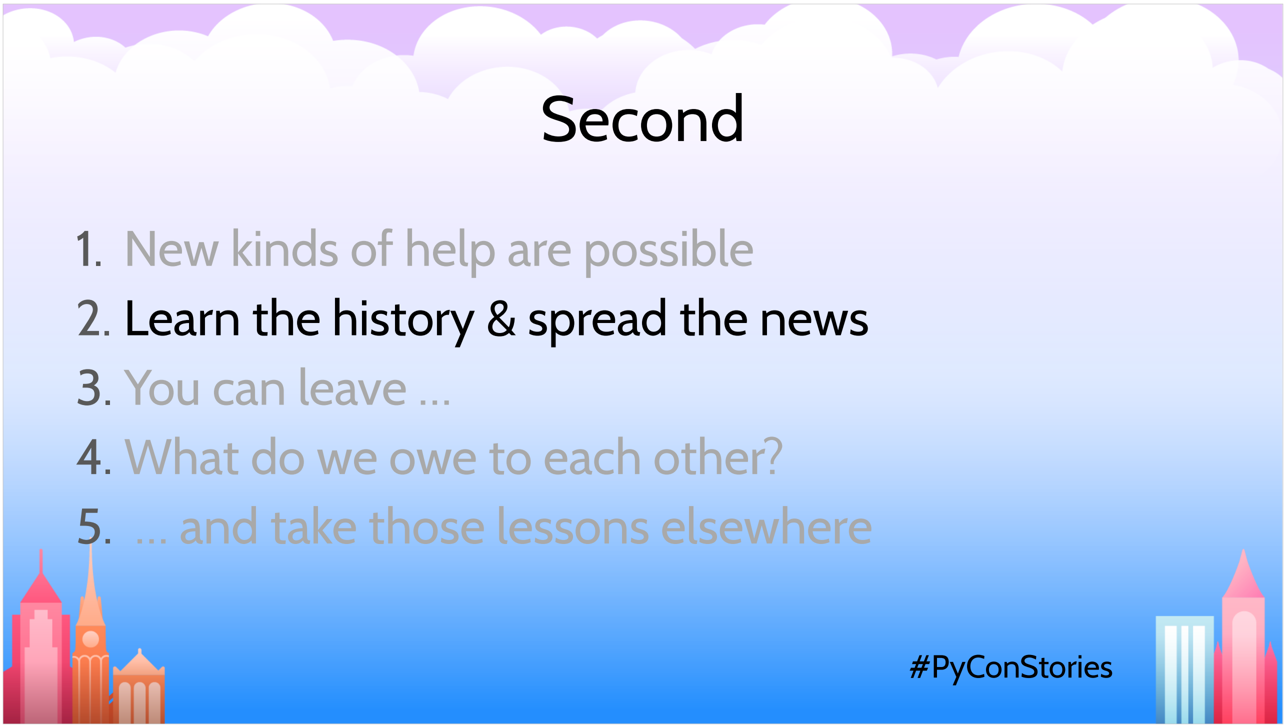
#PyConStories

Fun fact, I never ended up working on NumPy. [LAUGHTER]

And I think there's kind of a lesson there. What seems like failure can actually prepare the ground for success. One reason why we were successful in even that first Mozilla Open Source Support application: I knew something about the process because I had failed previously in applying for a MOSS grant for a different project!

I've been really inspired recently by Rebecca Solnit's book, *Hope in the Dark*, which is about activists who think, "Aaah, I have just failed. This thing that we're trying to do, we didn't do it." But actually, in the course of trying, they built foundational organizations, networks, knowledge, skills that turned out to be absolutely crucial in a future success that they just couldn't foresee in the moment.

Hope means you just don't know what's coming next.

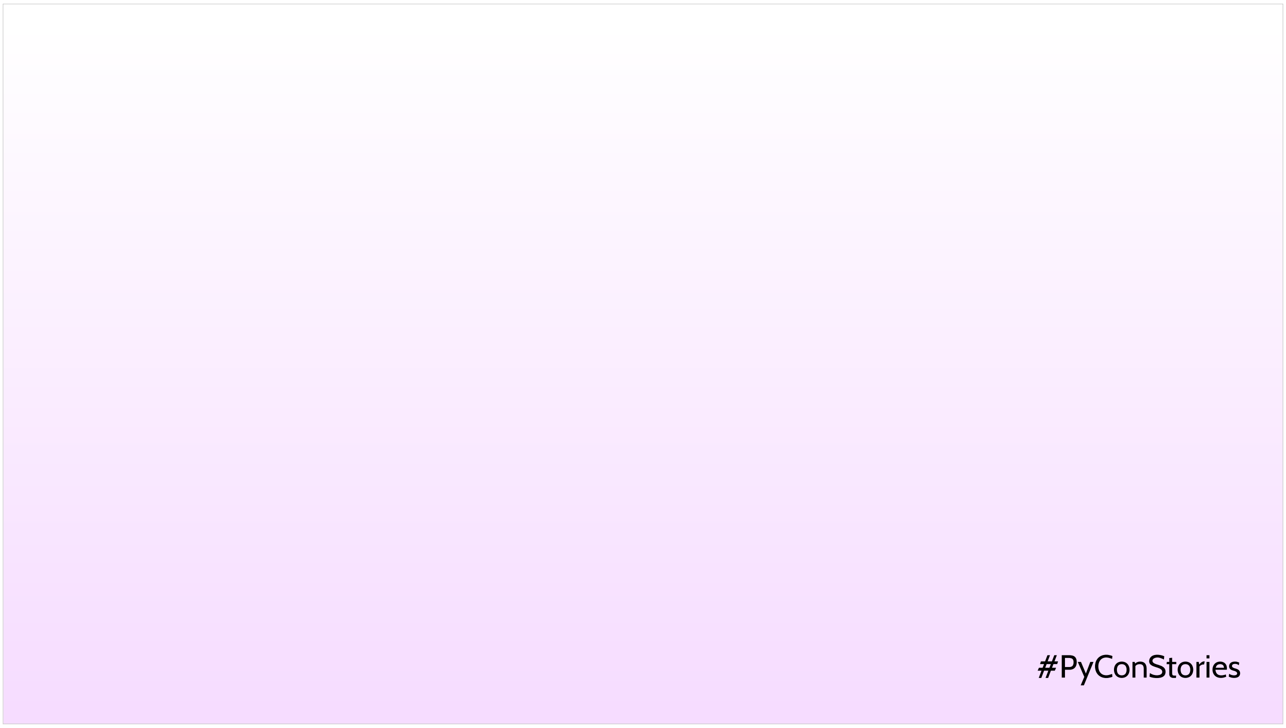


Second

1. New kinds of help are possible
2. Learn the history & spread the news
3. You can leave …
4. What do we owe to each other?
5. … and take those lessons elsewhere

#PyConStories

The second story I want to tell you is how I engaged with learning and teaching with Python packaging, because engaging with a big old infrastructure project involves learning its history and then spreading the news.



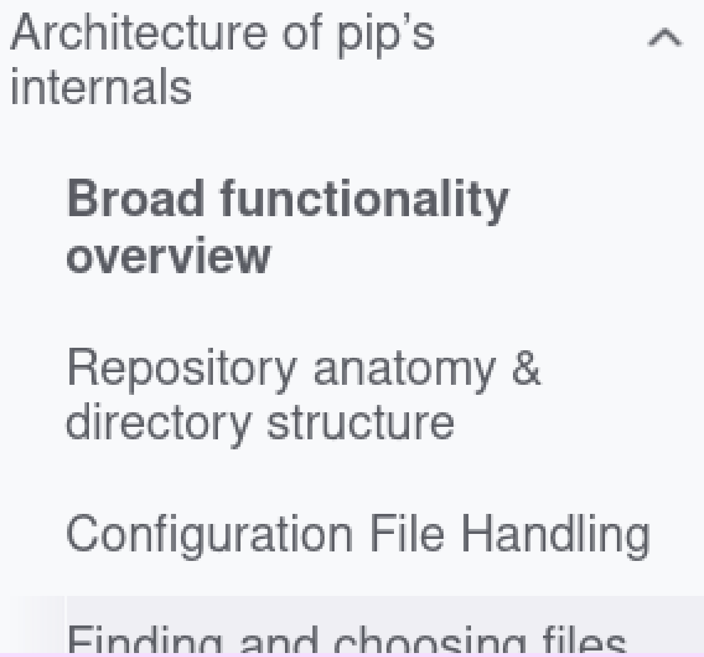
#PyConStories

Oh, again, like, there's so much confusion. I'm not going to try to get into the details of the internals. If you think dealing with packaging as a user is confusing, you ought to try learning how to maintain it. And, oh, if you **ever** start working on anything as complicated as Python packaging, on the maintainer level, you will never again use the phrase, "why don't you just?" [BIG LAUGHTER AND APPLAUSE]

Python packaging, that entire set of tools, mostly emerged through volunteers putting together spare hours here and there. It was not a top-down design process well-funded by the industry that depends on it. And as a result, it was iterative. It's been organic. There is a lot that makes sense, but that whole process doesn't lead to really consistent and easy-to-understand-from-the-outside architecture and interfaces.

And if you start engaging with an old, iteratively created project of that type, and you get confused because the anatomy of it feels undesigned, it's hard to understand, **you are the most normal person alive.**

Sometimes you have trouble understanding the system **because** there's something in it that doesn't make sense.

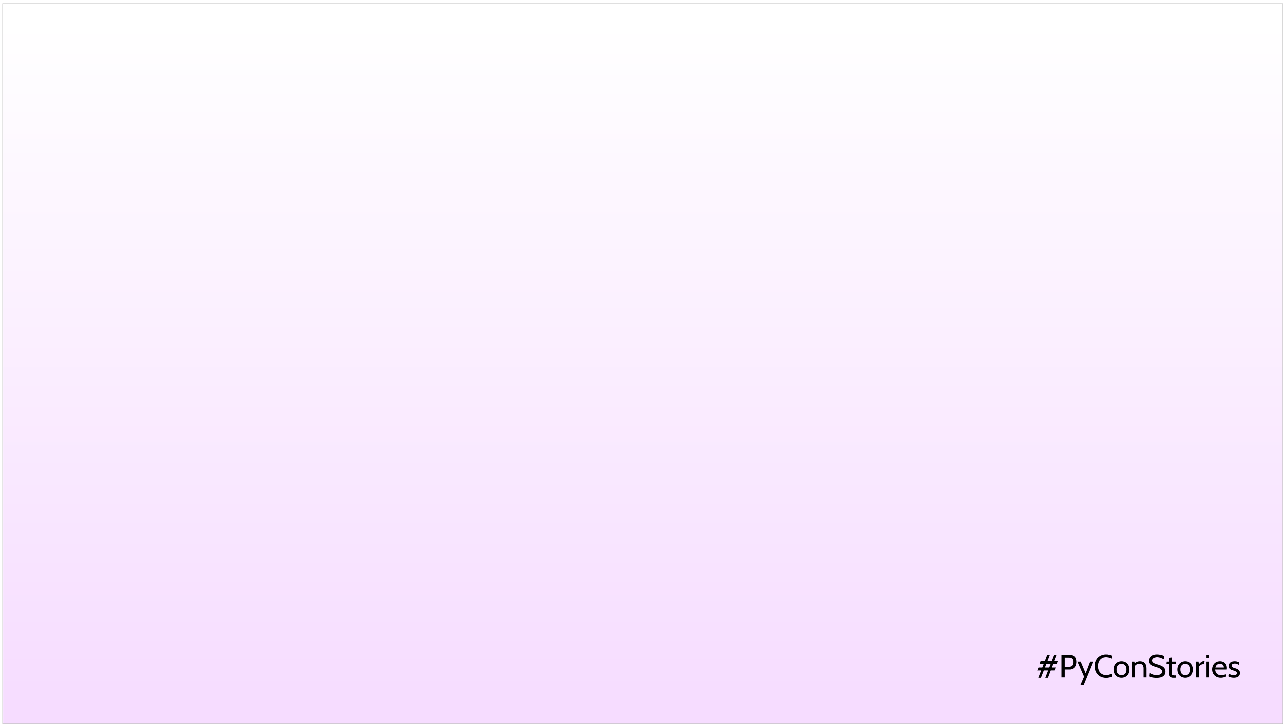

Architecture of pip’s internals
Broad functionality overview
Repository anatomy & directory structure
Configuration File Handling
Finding and choosing files

#PyConStories

I **used** my learner's mind. I asked questions and I wrote down the answers. And now we have better documentation. And it's going to be better for sustainability if the -- [PUT COIN IN SWEAR JAR] (Such a dirty mouth.)

One way to think about an open source project is that it's like a lab notebook, when you have documentation that you share together. The git commits, the comments, the manual, all of it, right? Everybody who's contributing needs to learn to **read** that shared lab notebook, and everyone needs to learn to **write** in it in a way that can be reused by others. I found that asking, "hey, why is this the way it is?" led to interesting architectural overviews, that I could then put into documentation, like this example from pip's documentation, now.

I highly recommend the book *Making Software, What Really Works and Why We Believe It*, edited by Greg Wilson and Andy Oram -- there's a link in the blog post -- which is a compendium of research in software engineering. And one of the findings mentioned in it is: the **number one thing** that new contributors coming onto a team want, that they can't find in existing documentation, is that **design rationale**. "Why is this bit designed the way it is? Was it intentional? Was it to meet a deadline? Was it an accident?"

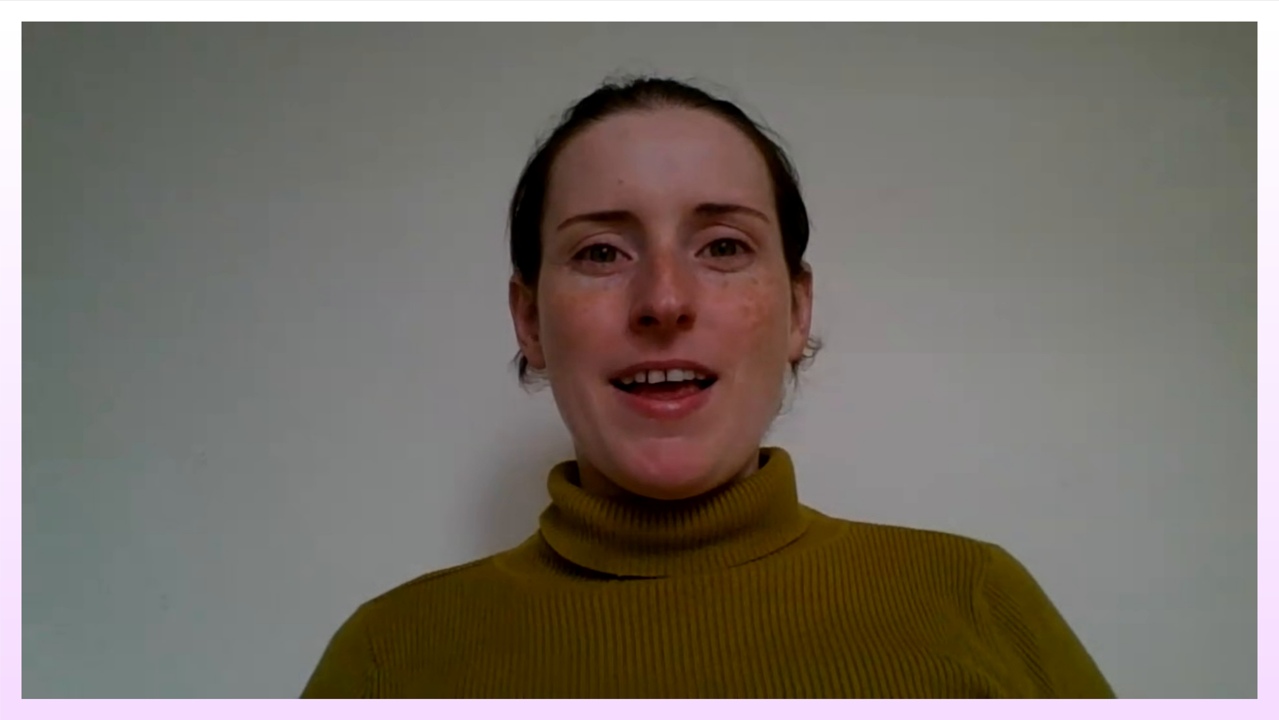


#PyConStories

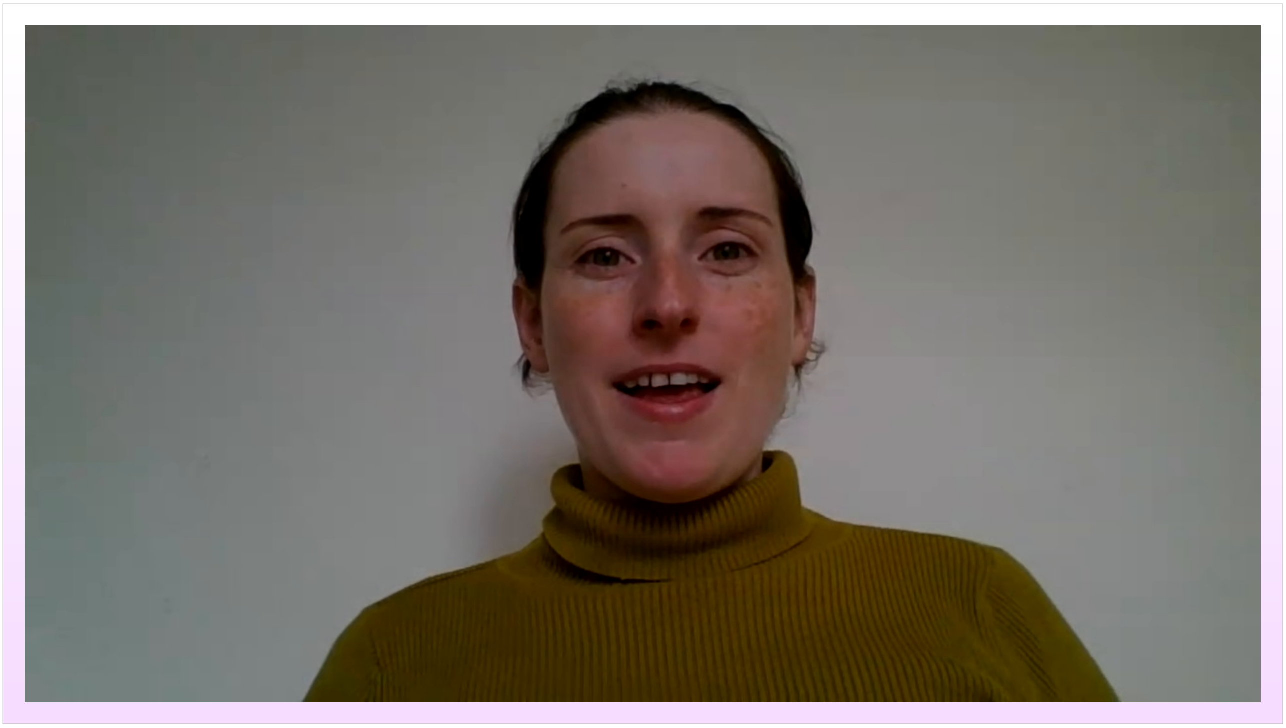
You can learn these things when you come into an existing project. You become an archaeologist, right? You contribute, you learn about the past. That's what **I** did.

And **then**, to help our users, I needed to help them learn about the present and the future. Which is especially hard to do when you're trying to do outreach about **infrastructure**, which, by **definition**, is stuff that a lot of people depend on who don't know they depend on it.

So, as I was working on, especially, PyPI and pip, I tried out a few different ways to spread that news. And one of them was this video, which I've just very lightly edited.

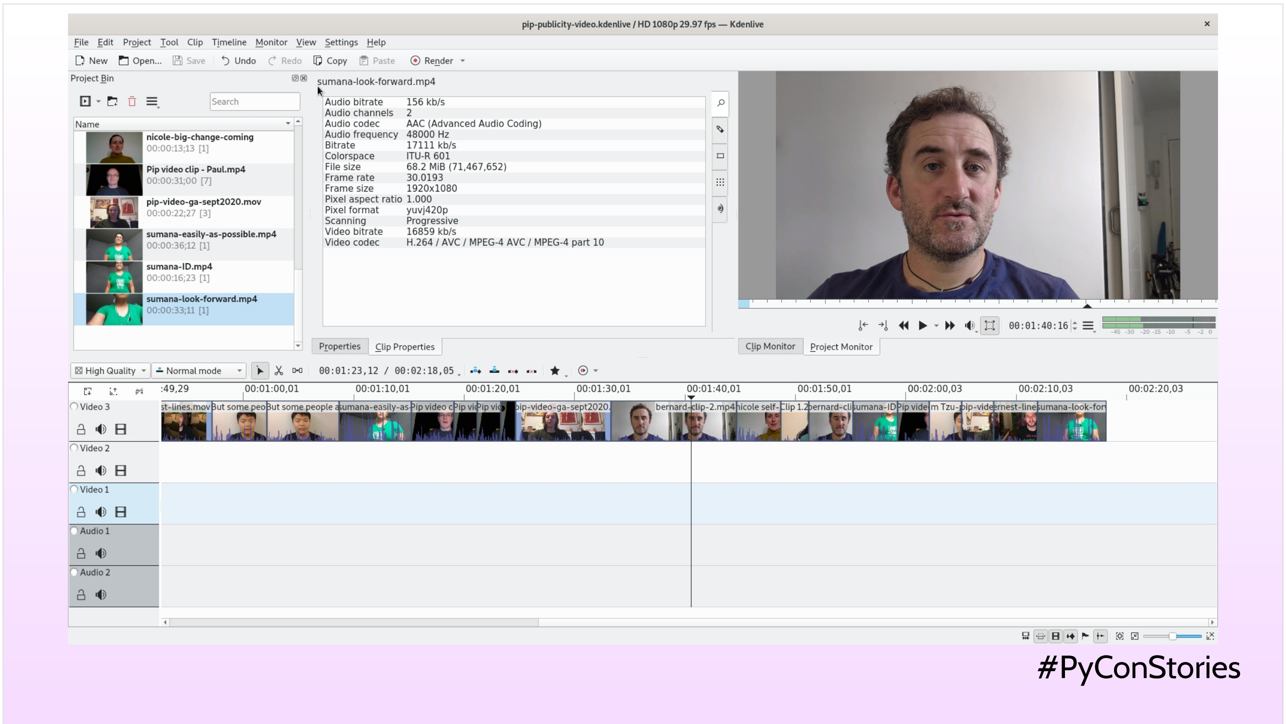


[PRE-RECORDED ViDEO PLAYS, SHOWING SEVERAL FACES WITH LIGHT MUSIC]
Nicole: There is a big change coming to pip in October. It'll be a great foundation for making it easier to deal with installing packages in Python.
Pradyun: Computers need to know the right order to install pieces of software ("to install x, you need to install y first").
Bernard: So, when Python programmers share software, like when they publish packages on PyPI, they have to precisely describe those installation prerequisites.
Tzu-ping: And then pip needs to navigate tricky situations when it gets conflicting instructions.
Pradyun: Up until now, pip's been **very** inconsistent in handling this stuff, which makes it easy for your Python environment to get messed up.
Paul: So we've been adding a proper dependency resolver to pip, so it's better at handling that tricky logic, and easier for you to use and troubleshoot.
Georgia: We're testing it in beta right now, and in pip 20.3, coming in October, the new behavior will be the default.
Ee: Once you're using the new resolver, pip is going to be more strict and more consistent. So things won't mysteriously break as much, and new features can be added that lots of people want.
Tzu-ping: But some people and some packages are going to run into problems in October. Because the transition to a stricter approach will break some workarounds that depend on inconsistencies.
Sumana: We **really** want this change to go as smoothly as possible, for you, **and** for our bugtracker.
Paul: So now's a good time for you to visit our migration guide, at the link below [https://pip.pypa.io], and try the beta of the new resolver. There's a link there for you to report feedback, good **and** bad….



...Georgia: And the **other** link [https://bit.ly/pip-ux-studies] is to sign up for our user experience studies. If you sign up then we can get your opinions in the future, with surveys and interviews, to help make Python packaging better.
Bernard: No matter whether you use Python every day or not, whether you're an open source maintainer or an open source software user, please sign up, because your opinion matters.
Nicole: I'm Nicole and I work on user experience for pip and Python packaging.
Pradyun: I'm Pradyun and I'm a pip maintainer.
Bernard: I'm Bernard and I work on the user experience of pip.
Sumana: I'm Sumana and I'm the project manager for pip.
Paul: I'm Paul and I'm a pip maintainer.
Tzu-ping: I'm Tzu-ping and I'm a contributor to pip.
Georgia: I'm Georgia and I work on pip's user experience.
Ee: I'm Ee and I work on the Python packaging infrastructure.
Sumana: We're some of the people who'll read your bug reports and your compliments. We look forward to hearing from you.

[BIG APPLAUSE]



pip-publicity-video.kdenlive / HD 1080p 29.97 fps — Kdenlive

Project Bin:

- nicole-big-change-coming 00:00:13;13 [1]
- Pip video clip - Paul.mp4 00:00:31;00 [7]
- pip-video-ga-sept2020.mov 00:00:22;27 [3]
- sumana-easily-as-possible.mp4 00:00:36;12 [1]
- sumana-ID.mp4 00:00:16;23 [1]
- sumana-look-forward.mp4 00:00:33;11 [1]

sumana-look-forward.mp4

Audio bitrate	156 kb/s
Audio channels	2
Audio codec	AAC (Advanced Audio Coding)
Audio frequency	48000 Hz
Bitrate	17111 kb/s
Colorspace	ITU-R 601
File size	68.2 MiB (71,467,652)
Frame rate	30.0193
Frame size	1920x1080
Pixel aspect ratio	1.000
Pixel format	yuvj420p
Scanning	Progressive
Video bitrate	16859 kb/s
Video codec	H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10

00:01:40;16

00:01:23,12 / 00:02:18,05

#PyConStories

[BIG APPLAUSE]

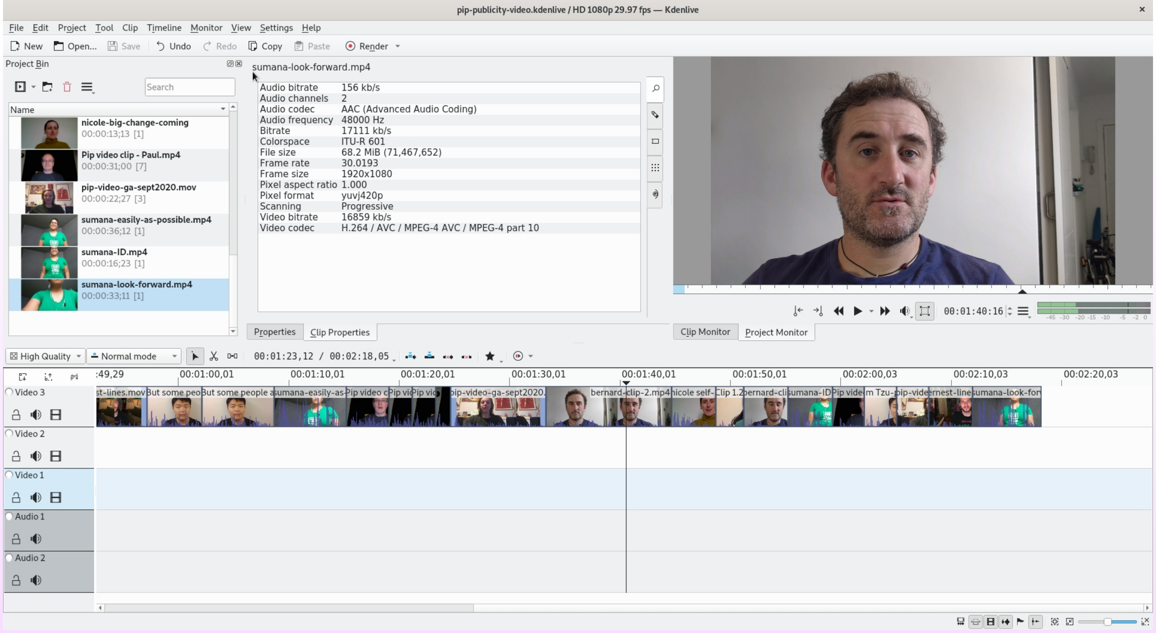
Thank you. Glad you liked it.

I want to point out two things about this video we made.

One, what it's trying to do, that's pretty clear, pretty obvious on the surface: boil down what people need to know and pass it along. "Here's some stuff that's going to happen to something you depend on. Heads up, here's how to give us some feedback about it." And it's something that we were trying to get out to people who weren't going to go read the official Python blog. Right? People who don't even think of themselves as pip users the way... I drink **water**, but I don't think of myself as a **water user**. I'm not going to the water Discord, [LAUGHTER] catching up on the latest water release notes. [LAUGHTER]

Infrastructure is, often, invisible either just because of our perception, and sometimes in the case of digital infrastructure, often almost literally. And Mel Chua and Stephen Jacobs did some research specifically about PyPI and... sustainability. [DROPS COIN IN JAR]

And talking about a particular interesting tension: "Expectations of whether communication from upstream to downstream should happen. The expectation that users should be able to maintain an uninterrupted focus on their own work (which is encapsulation), versus the expectation that users should have an ongoing awareness of happenings upstream (which is not-encapsulation)."...



pip-publicity-video.kdenlive / HD 1080p 29.97 fps — Kdenlive
File Edit Project Tool Clip Timeline Monitor View Settings Help
New Open... Save Undo Redo Copy Paste Render
Project Bin
Search
Name
Properties Clip Properties
Clip Monitor Project Monitor
High Quality
Normal mode
Video 3
Video 2
Video 1
Audio 1
Audio 2

nicole-big-change-coming
00:00:13;13 [1]

Pip video clip - Paul.mp4
00:00:31;00 [7]

pip-video-ga-sept2020.mov
00:00:22;27 [3]

sumana-easily-as-possible.mp4
00:00:36;12 [1]

sumana-ID.mp4
00:00:16;23 [1]

sumana-look-forward.mp4
00:00:33;11 [1]

sumana-look-forward.mp4

Audio bitrate	156 kb/s
Audio channels	2
Audio codec	AAC (Advanced Audio Coding)
Audio frequency	48000 Hz
Bitrate	17111 kb/s
Colorspace	ITU-R 601
File size	68.2 MiB (71,467,652)
Frame rate	30.0193
Frame size	1920x1080
Pixel aspect ratio	1.000
Pixel format	yuvj420p
Scanning	Progressive
Video bitrate	16859 kb/s
Video codec	H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10

00:01:40:16

00:01:23,12 / 00:02:18,05

#PyConStories

...Right? We live in that paradox. And it's **especially** magnified, given that there is no one place that I could communicate out to everyone who needed to hear about this change. In the US, if I were the water company, a utility like that, I'd at least have some **billing** addresses. But I don't have that.

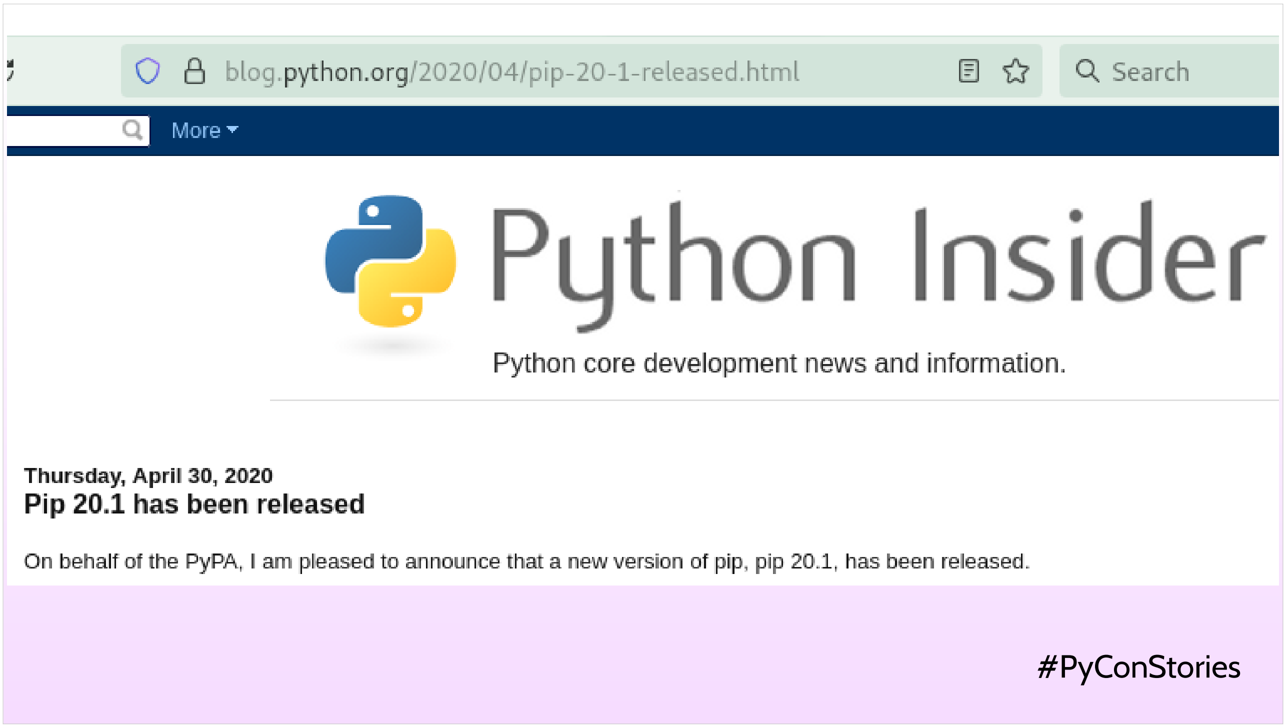
There is no one Python community. There's an archipelago. Without a map. So trying to get this video out was a way to try and get users who live in some different spots, right? In that archipelago.

And then there's the **subtext** of this video, which is: "don't yell at us, [LAUGHTER] if you get mad."

It is **absolutely reasonable** for users to get frustrated when institutions foist changes on us, without any kind of recourse. When we are just at the whim of people who hide, and there's no accountability.

I didn't want us to do that.

So this video tries to say, "We really want your feedback. And when you give it to us, please treat us like people, because we are. We're people just like you, and we're all in this together." [BIG APPLAUSE]


blog.python.org/2020/04/pip-20-1-released.html
Search
More
Python Insider
Python core development news and information.

Thursday, April 30, 2020

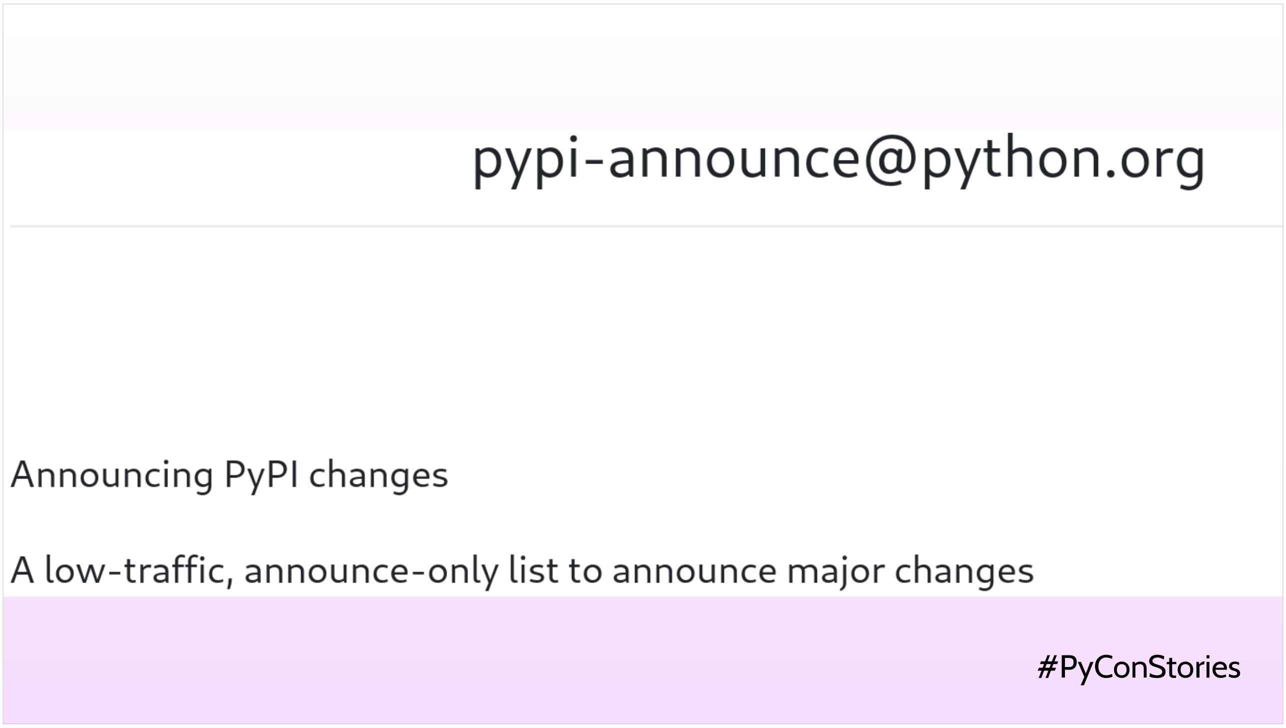
Pip 20.1 has been released

On behalf of the PyPA, I am pleased to announce that a new version of pip, pip 20.1, has been released.

#PyConStories

So it was a broad-spectrum approach, right?

Okay, put it on the official Python blog...



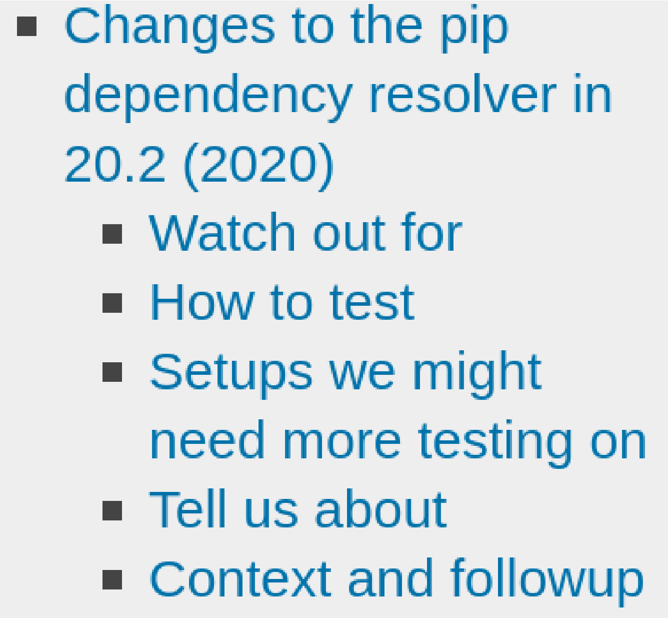
pypi-announce@python.org

Announcing PyPI changes

A low-traffic, announce-only list to announce major changes

#PyConStories

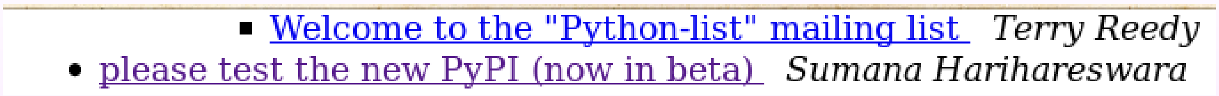
...and our low-traffic announcement mailing list...



- Changes to the pip dependency resolver in 20.2 (2020)
    - Watch out for
    - How to test
    - Setups we might need more testing on
    - Tell us about
    - Context and followup

#PyConStories

...write stuff in our documentation...

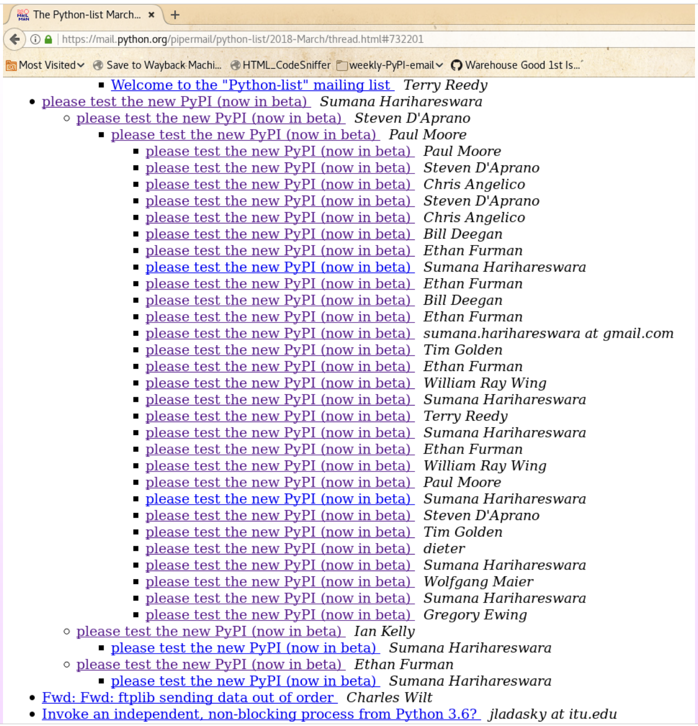


▪ Welcome to the "Python-list" mailing list *Terry Reedy*

- please test the new PyPI (now in beta) *Sumana Harihareswara*

#PyConStories

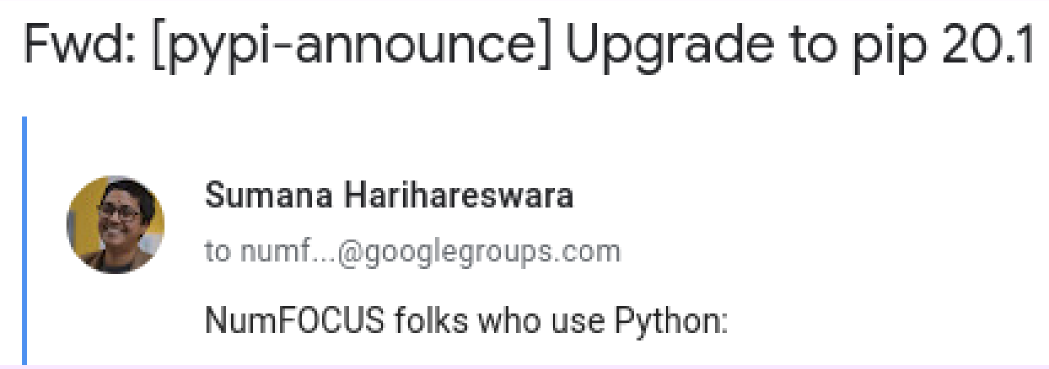
...post to the main Python mailing list...



        - Welcome to the "Python-list" mailing list *Terry Reedy*
- please test the new PyPI (now in beta) *Sumana Harihareswara*
    - please test the new PyPI (now in beta) *Steven D'Aprano*
        - please test the new PyPI (now in beta) *Paul Moore*
            - please test the new PyPI (now in beta) *Paul Moore*
            - please test the new PyPI (now in beta) *Steven D'Aprano*
            - please test the new PyPI (now in beta) *Chris Angelico*
            - please test the new PyPI (now in beta) *Steven D'Aprano*
            - please test the new PyPI (now in beta) *Chris Angelico*
            - please test the new PyPI (now in beta) *Bill Deegan*
            - please test the new PyPI (now in beta) *Ethan Furman*
            - please test the new PyPI (now in beta) *Sumana Harihareswara*
            - please test the new PyPI (now in beta) *Ethan Furman*
            - please test the new PyPI (now in beta) *Bill Deegan*
            - please test the new PyPI (now in beta) *Ethan Furman*
            - please test the new PyPI (now in beta) *sumana.harihareswara at gmail.com*
            - please test the new PyPI (now in beta) *Tim Golden*
            - please test the new PyPI (now in beta) *Ethan Furman*
            - please test the new PyPI (now in beta) *William Ray Wing*
            - please test the new PyPI (now in beta) *Sumana Harihareswara*
            - please test the new PyPI (now in beta) *Terry Reedy*
            - please test the new PyPI (now in beta) *Sumana Harihareswara*
            - please test the new PyPI (now in beta) *Ethan Furman*
            - please test the new PyPI (now in beta) *William Ray Wing*
            - please test the new PyPI (now in beta) *Paul Moore*
            - please test the new PyPI (now in beta) *Sumana Harihareswara*
            - please test the new PyPI (now in beta) *Steven D'Aprano*
            - please test the new PyPI (now in beta) *Tim Golden*
            - please test the new PyPI (now in beta) *dieter*
            - please test the new PyPI (now in beta) *Sumana Harihareswara*
            - please test the new PyPI (now in beta) *Wolfgang Maier*
            - please test the new PyPI (now in beta) *Sumana Harihareswara*
            - please test the new PyPI (now in beta) *Gregory Ewing*
    - please test the new PyPI (now in beta) *Ian Kelly*
        - please test the new PyPI (now in beta) *Sumana Harihareswara*
    - please test the new PyPI (now in beta) *Ethan Furman*
        - please test the new PyPI (now in beta) *Sumana Harihareswara*
- Fwd: Fwd: ftplib sending data out of order *Charles Wilt*
- Invoke an independent, non-blocking process from Python 3.6? *jladasky at itu.edu*

#PyConStories

...and answer questions there....



Fwd: [pypi-announce] Upgrade to pip 20.1

Sumana Harihareswara

to numf...@googlegroups.com

NumFOCUS folks who use Python:

#PyConStories

...**And:**

I searched GitHub for projects that had experienced a particular kind of problem with our project, and then told them, "hey, please check out this beta."

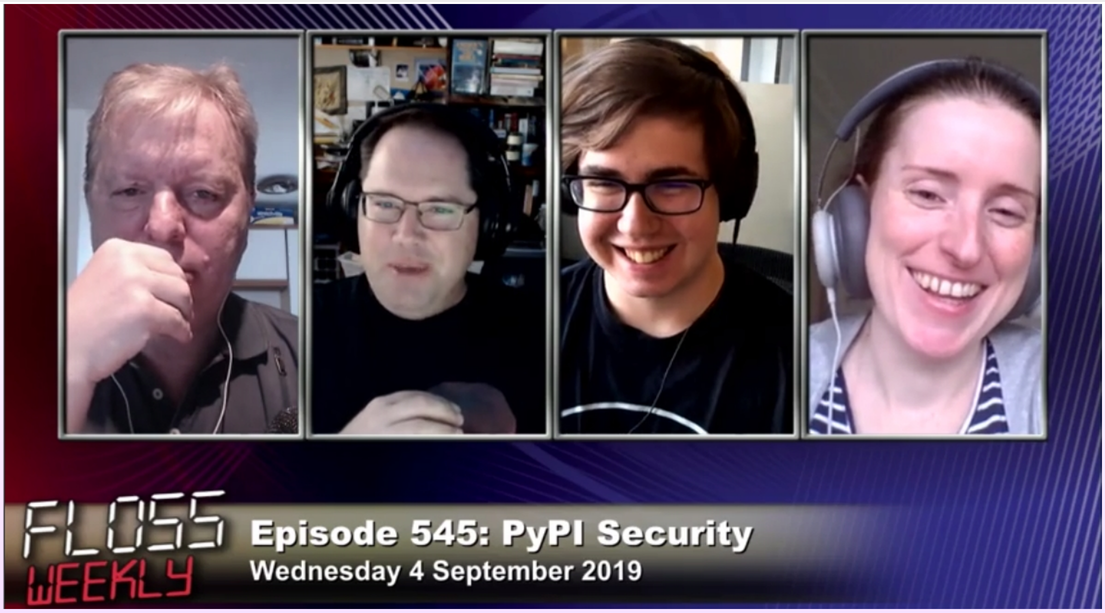
I wrote an article in *Linux Weekly News*.

I asked newsletters that are more Python-specific to spread the word.

I said things on Twitter, on a chat channel. I set up an open chat hour.

I tried to reach out to language-specific, you know, the **Spanish**-speaking users, to try and get the word out that way.

And I wrote to mailing lists, and I had conference calls with data journalists, teachers, research scientists, those kinds of groups who **use** Python, **would** be affected by this, but do not **conceptualize** themselves as Python users and seek information directly that way.

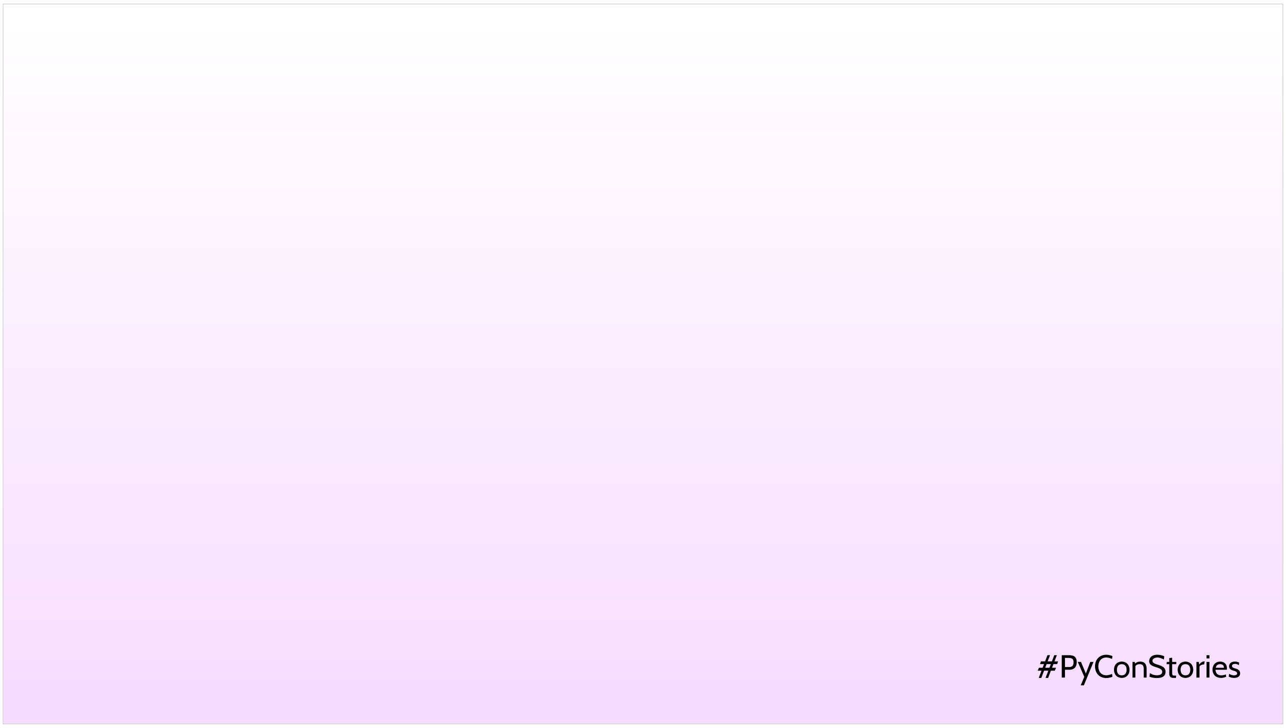


#PyConStories

I reached out to Python-**specific** podcasts, but -- you know, *Talk Python*, *Python Bytes*, *Podcast.__init__* -- but also, *FLOSS Weekly*, and other podcasts that were more generally about free and open source software.

I could have used TikTok, maybe. An actual **ad budget**, even, would have been useful.

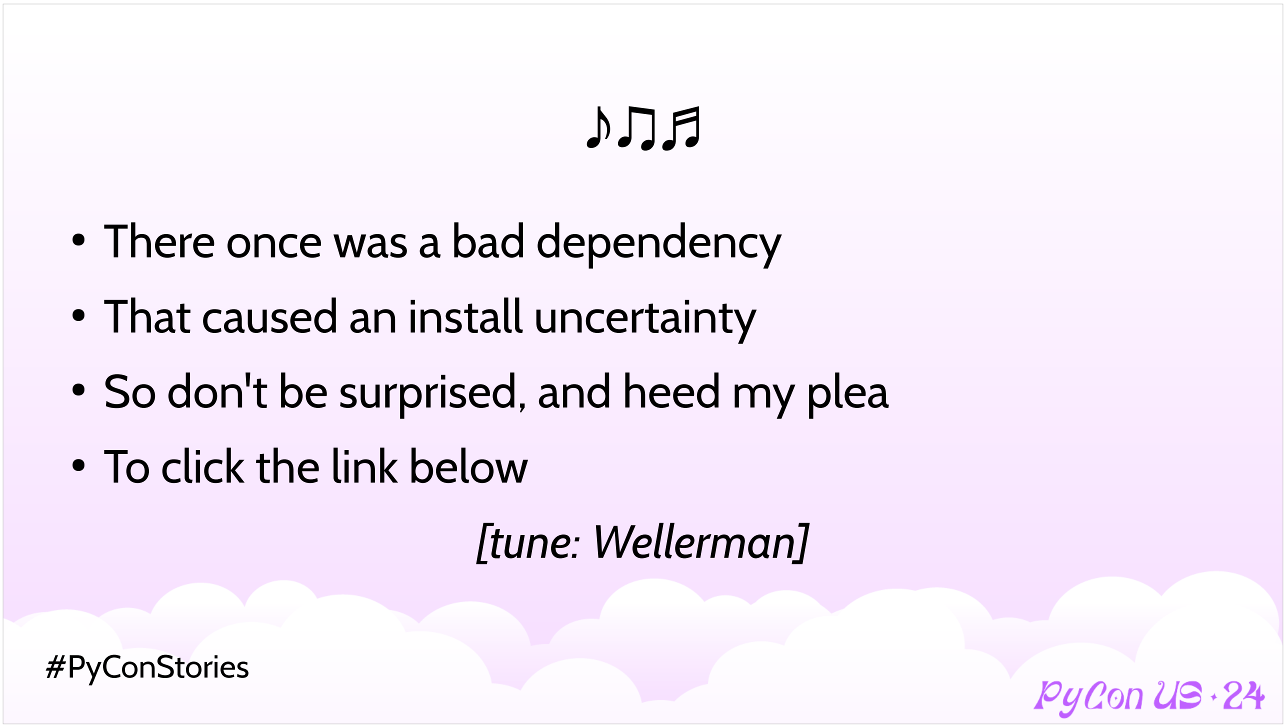
Because it is great to be where your audience **is** instead of blaming them for not being where **you** are.



#PyConStories

Heh – I mean, it was 2020. Maybe I should have done, like, a sea shanty. [LAUGHTER]

Because -- for those of you who were not hanging out on English-language social media, in 2020 there was a song called "The Wellerman" that went around on English-language social media.



♪♫♬

- There once was a bad dependency
- That caused an install uncertainty
- So don't be surprised, and heed my plea
- To click the link below

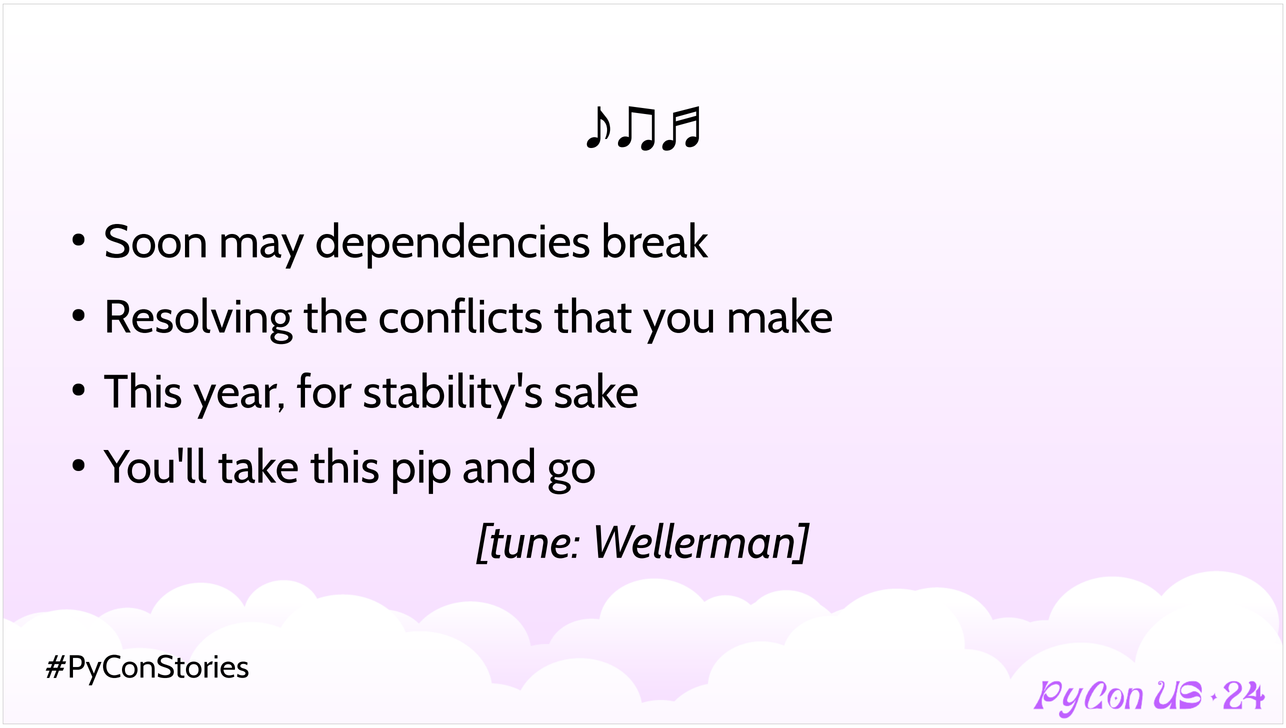
[tune: Wellerman]

#PyConStories

PyCon US · 24

[I BEGIN SINGING AND PEOPLE START LAUGHING]

There once was a bad dependency
that caused an install uncertainty.
So don't be surprised and heed my plea
to click the link below.



♪♫♬

- Soon may dependencies break
- Resolving the conflicts that you make
- This year, for stability's sake
- You'll take this pip and go

[tune: Wellerman]

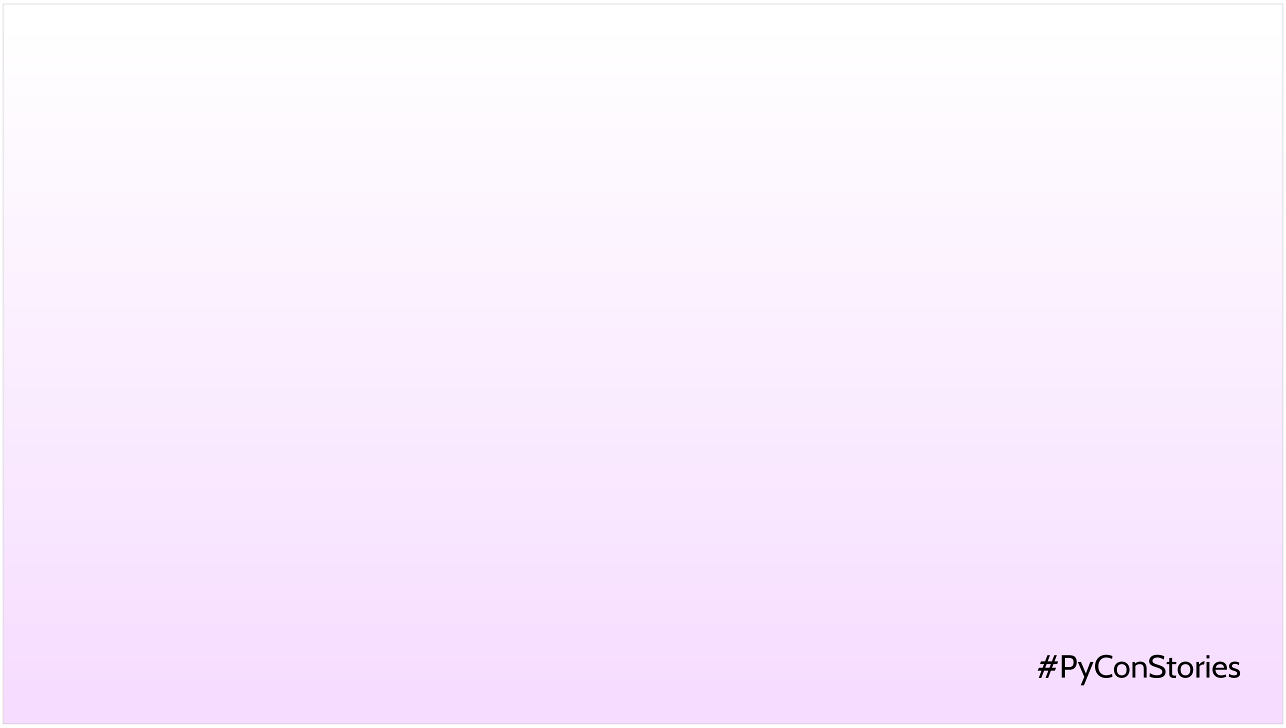
#PyConStories

PyCon US · 24

[GUY WITH BASS VOICE JOINS IN; PEOPLE ARE CLAPPING ALONG]

Soon may dependencies break,
resolving the conflicts that you make.
This year for stability's sake,
you'll take this pip and go.

[END OF SONG; BIG APPLAUSE]

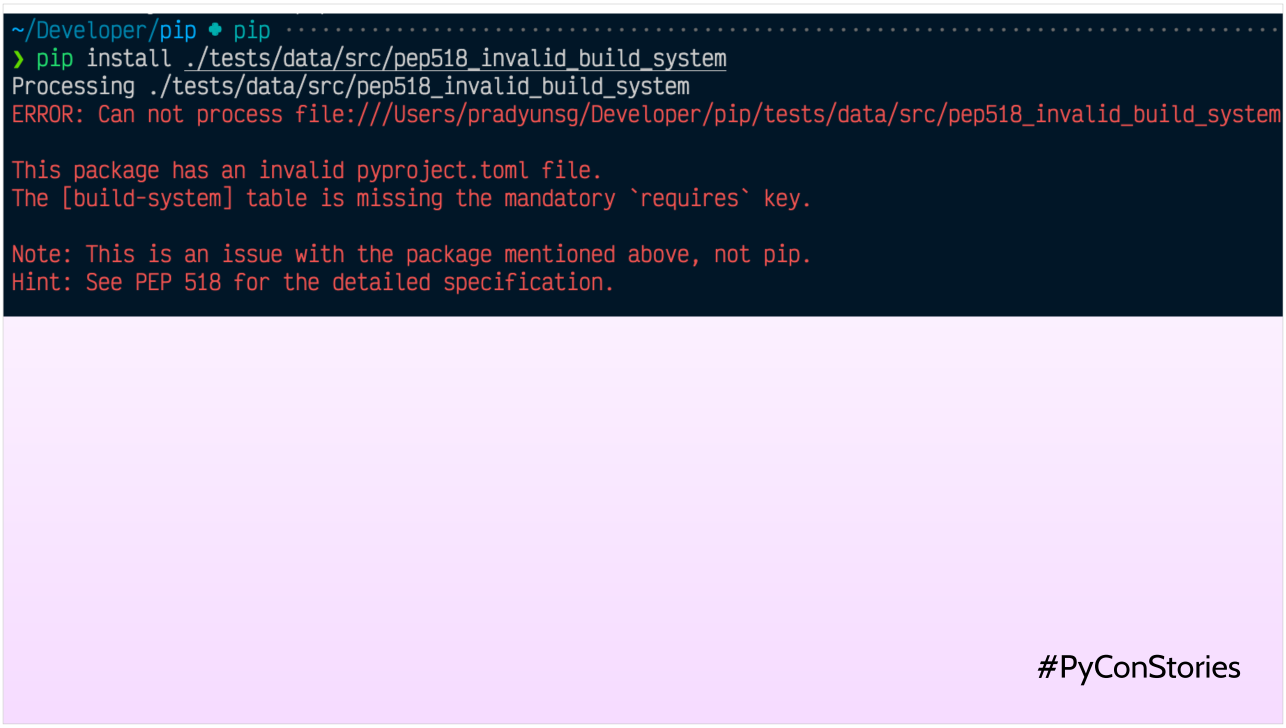


#PyConStories

(That probably would have ballooned the executable. [LAUGHTER] It's a good thing we didn't do it.)

A thing that **does** work is error messages and terminal output. Even on the command line, right? You have a user experience that includes error messages, and sometimes error messages are the only kind of public service announcement that you can be guaranteed at least some users will read. So, if you have a user, that means you **have** a user interface and you can improve it.

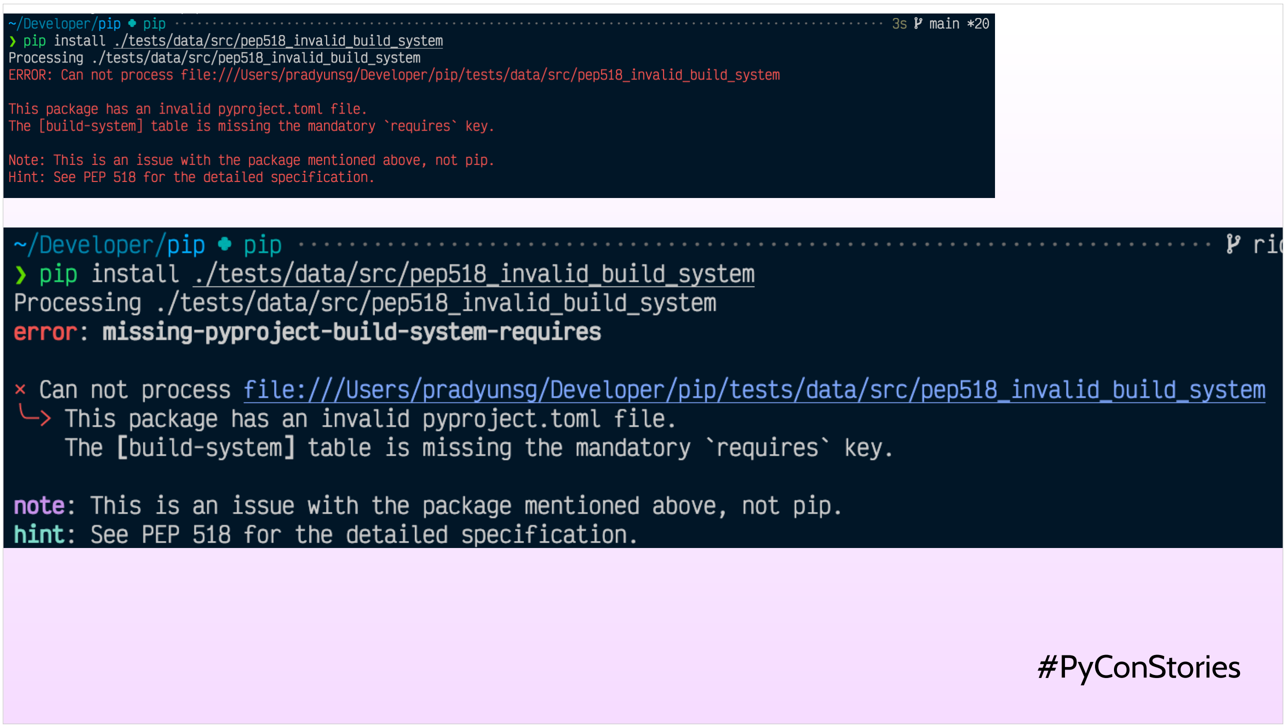
And one of the things I'm **most** proud of, in our work on the pip resolver in 2020, is that I helped make sure there was budget for user experience research people, who could actually do surveys, user tests, interviews, talk with users and help figure out: "What is their mental model in a difficult moment of trying to resolve a dependency conflict? What could we do to test, A/B test, different error messages in terms of color, length, formatting, hyperlinks?"



```
~/Developer/pip ✚ pip ···························································
❯ pip install ./tests/data/src/pep518_invalid_build_system
Processing ./tests/data/src/pep518_invalid_build_system
ERROR: Can not process file:///Users/pradyunsg/Developer/pip/tests/data/src/pep518_invalid_build_system

This package has an invalid pyproject.toml file.
The [build-system] table is missing the mandatory `requires` key.

Note: This is an issue with the package mentioned above, not pip.
Hint: See PEP 518 for the detailed specification.
```

#PyConStories

Here is a before and after, for instance, from a big block of red...

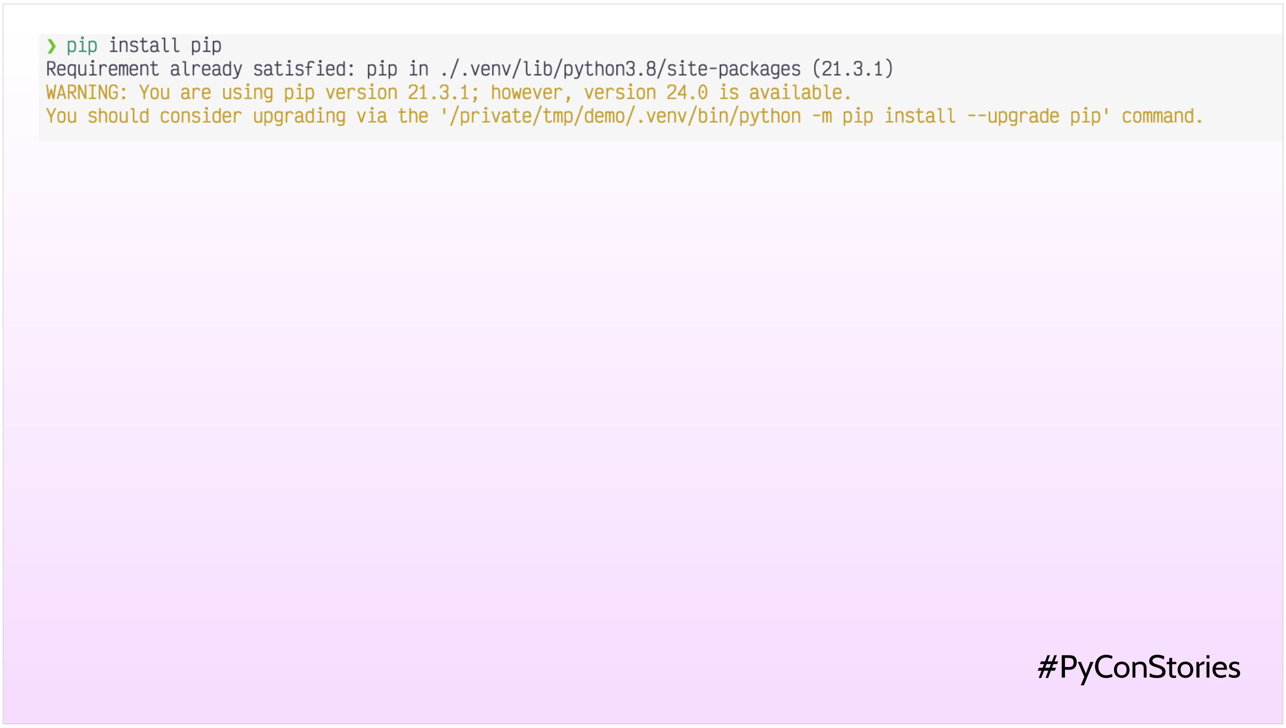


```
~/Developer/pip ◆ pip ·············································································· 3s ᚠ main *20
❯ pip install ./tests/data/src/pep518_invalid_build_system
Processing ./tests/data/src/pep518_invalid_build_system
ERROR: Can not process file:///Users/pradyunsg/Developer/pip/tests/data/src/pep518_invalid_build_system

This package has an invalid pyproject.toml file.
The [build-system] table is missing the mandatory `requires` key.

Note: This is an issue with the package mentioned above, not pip.
Hint: See PEP 518 for the detailed specification.
```

```
~/Developer/pip ◆ pip ············································ ᚠ ri
❯ pip install ./tests/data/src/pep518_invalid_build_system
Processing ./tests/data/src/pep518_invalid_build_system
error: missing-pyproject-build-system-requires

× Can not process file:///Users/pradyunsg/Developer/pip/tests/data/src/pep518_invalid_build_system
╰─> This package has an invalid pyproject.toml file.
    The [build-system] table is missing the mandatory `requires` key.

note: This is an issue with the package mentioned above, not pip.
hint: See PEP 518 for the detailed specification.
```

#PyConStories

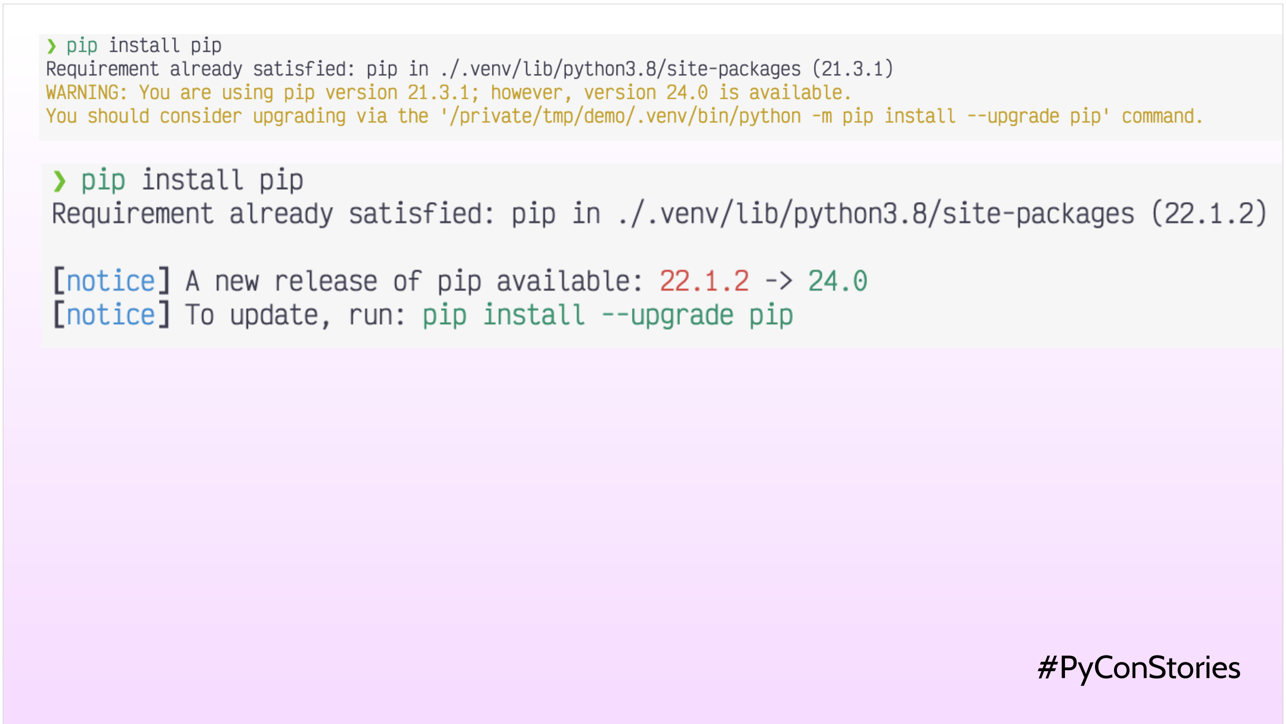
... to something that's formatted so you can actually understand what needs fixing.



```
❯ pip install pip
Requirement already satisfied: pip in ./.venv/lib/python3.8/site-packages (21.3.1)
WARNING: You are using pip version 21.3.1; however, version 24.0 is available.
You should consider upgrading via the '/private/tmp/demo/.venv/bin/python -m pip install --upgrade pip' command.
```

#PyConStories

Here's another big block of text that's the same color...

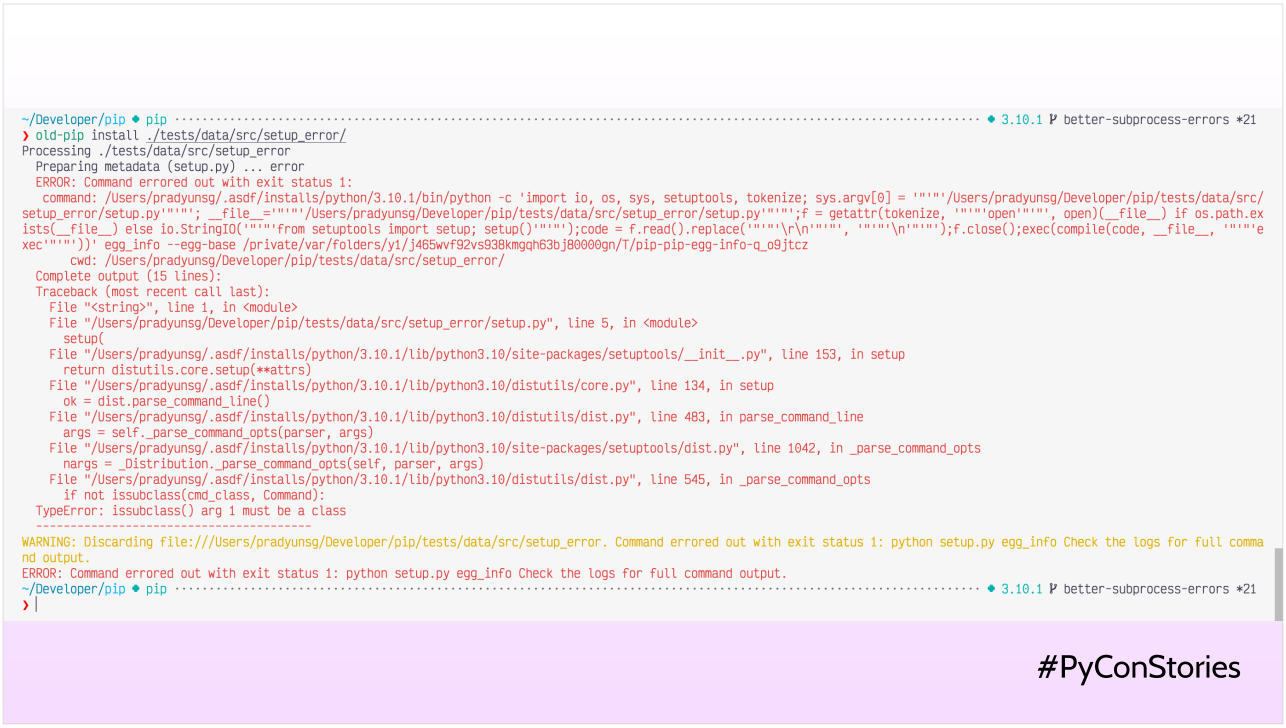


```
❯ pip install pip
Requirement already satisfied: pip in ./.venv/lib/python3.8/site-packages (21.3.1)
WARNING: You are using pip version 21.3.1; however, version 24.0 is available.
You should consider upgrading via the '/private/tmp/demo/.venv/bin/python -m pip install --upgrade pip' command.
```

```
❯ pip install pip
Requirement already satisfied: pip in ./.venv/lib/python3.8/site-packages (22.1.2)

[notice] A new release of pip available: 22.1.2 -> 24.0
[notice] To update, run: pip install --upgrade pip
```

#PyConStories

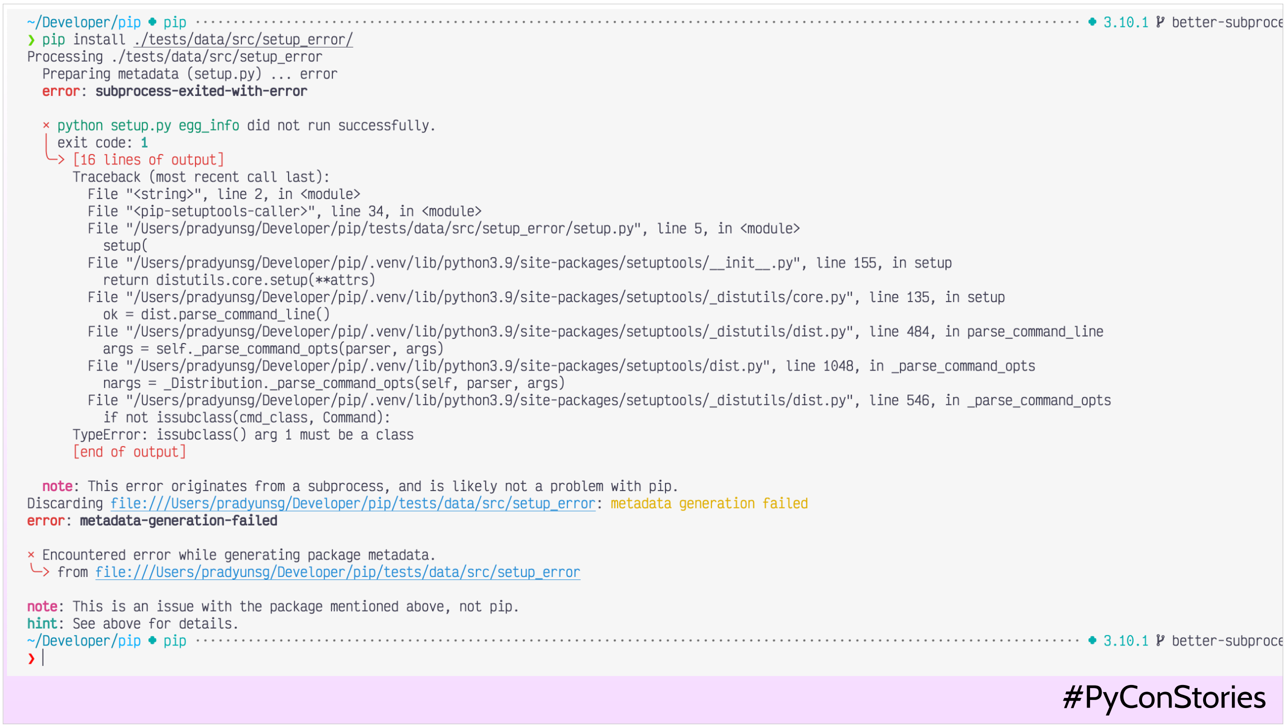
...less text, more pointed.



```
~/Developer/pip ◆ pip ·························································································· ◆ 3.10.1 ⎇ better-subprocess-errors *21
❯ old-pip install ./tests/data/src/setup_error/
Processing ./tests/data/src/setup_error
  Preparing metadata (setup.py) ... error
  ERROR: Command errored out with exit status 1:
   command: /Users/pradyunsg/.asdf/installs/python/3.10.1/bin/python -c 'import io, os, sys, setuptools, tokenize; sys.argv[0] = '"'"'/Users/pradyunsg/Developer/pip/tests/data/src/
setup_error/setup.py'"'"'; __file__='"'"'/Users/pradyunsg/Developer/pip/tests/data/src/setup_error/setup.py'"'"';f = getattr(tokenize, '"'"'open'"'"', open)(__file__) if os.path.ex
ists(__file__) else io.StringIO('"'"'from setuptools import setup; setup()'"'"');code = f.read().replace('"'"'\r\n'"'"', '"'"'\n'"'"');f.close();exec(compile(code, __file__, '"'"'e
xec'"'"'))' egg_info --egg-base /private/var/folders/y1/j465wvf92vs938kmgqh63bj80000gn/T/pip-pip-egg-info-q_o9jtcz
       cwd: /Users/pradyunsg/Developer/pip/tests/data/src/setup_error/
  Complete output (15 lines):
  Traceback (most recent call last):
    File "<string>", line 1, in <module>
    File "/Users/pradyunsg/Developer/pip/tests/data/src/setup_error/setup.py", line 5, in <module>
      setup(
    File "/Users/pradyunsg/.asdf/installs/python/3.10.1/lib/python3.10/site-packages/setuptools/__init__.py", line 153, in setup
      return distutils.core.setup(**attrs)
    File "/Users/pradyunsg/.asdf/installs/python/3.10.1/lib/python3.10/distutils/core.py", line 134, in setup
      ok = dist.parse_command_line()
    File "/Users/pradyunsg/.asdf/installs/python/3.10.1/lib/python3.10/distutils/dist.py", line 483, in parse_command_line
      args = self._parse_command_opts(parser, args)
    File "/Users/pradyunsg/.asdf/installs/python/3.10.1/lib/python3.10/site-packages/setuptools/dist.py", line 1042, in _parse_command_opts
      nargs = _Distribution._parse_command_opts(self, parser, args)
    File "/Users/pradyunsg/.asdf/installs/python/3.10.1/lib/python3.10/distutils/dist.py", line 545, in _parse_command_opts
      if not issubclass(cmd_class, Command):
  TypeError: issubclass() arg 1 must be a class
  ----------------------------------------
WARNING: Discarding file:///Users/pradyunsg/Developer/pip/tests/data/src/setup_error. Command errored out with exit status 1: python setup.py egg_info Check the logs for full comma
nd output.
ERROR: Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.
~/Developer/pip ◆ pip ·························································································· ◆ 3.10.1 ⎇ better-subprocess-errors *21
❯ |
```

#PyConStories

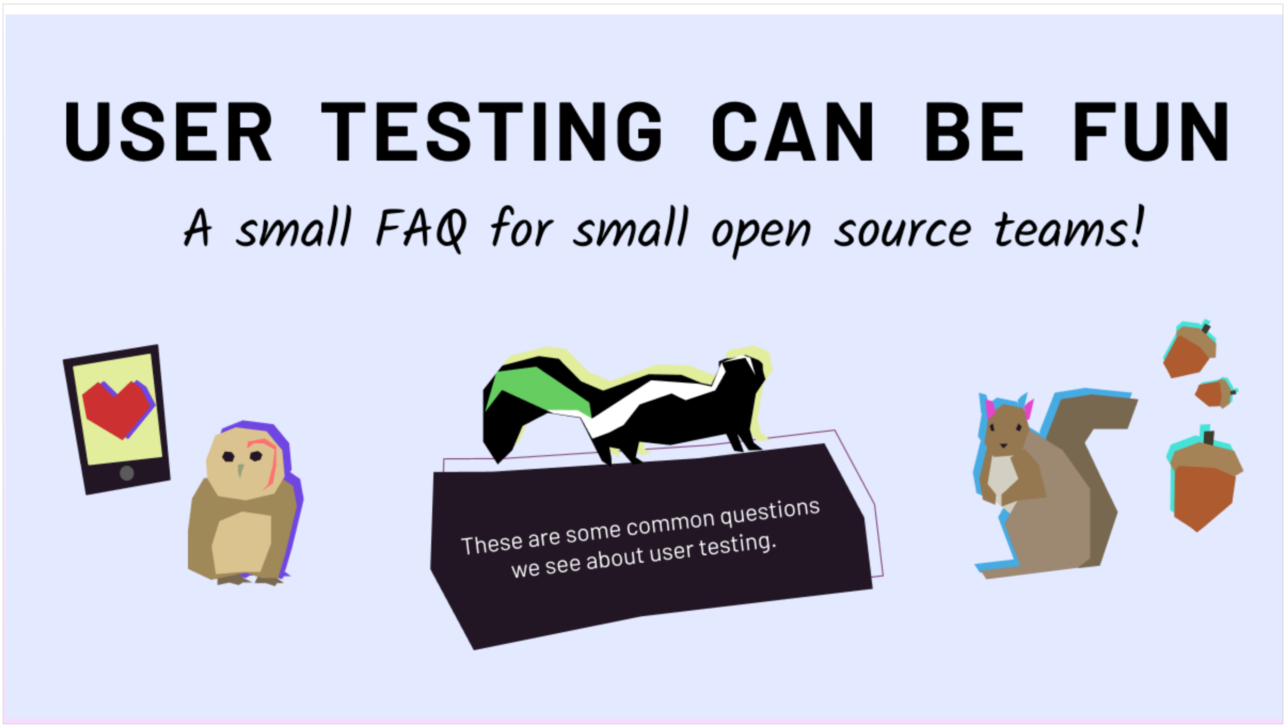
Here's a big red stack trace....



```
~/Developer/pip ◆ pip ·························································································· ◆ 3.10.1 better-subproce
❯ pip install ./tests/data/src/setup_error/
Processing ./tests/data/src/setup_error
  Preparing metadata (setup.py) ... error
  error: subprocess-exited-with-error

  × python setup.py egg_info did not run successfully.
  │ exit code: 1
  ╰─> [16 lines of output]
      Traceback (most recent call last):
        File "<string>", line 2, in <module>
        File "<pip-setuptools-caller>", line 34, in <module>
        File "/Users/pradyunsg/Developer/pip/tests/data/src/setup_error/setup.py", line 5, in <module>
          setup(
        File "/Users/pradyunsg/Developer/pip/.venv/lib/python3.9/site-packages/setuptools/__init__.py", line 155, in setup
          return distutils.core.setup(**attrs)
        File "/Users/pradyunsg/Developer/pip/.venv/lib/python3.9/site-packages/setuptools/_distutils/core.py", line 135, in setup
          ok = dist.parse_command_line()
        File "/Users/pradyunsg/Developer/pip/.venv/lib/python3.9/site-packages/setuptools/_distutils/dist.py", line 484, in parse_command_line
          args = self._parse_command_opts(parser, args)
        File "/Users/pradyunsg/Developer/pip/.venv/lib/python3.9/site-packages/setuptools/dist.py", line 1048, in _parse_command_opts
          nargs = _Distribution._parse_command_opts(self, parser, args)
        File "/Users/pradyunsg/Developer/pip/.venv/lib/python3.9/site-packages/setuptools/_distutils/dist.py", line 546, in _parse_command_opts
          if not issubclass(cmd_class, Command):
      TypeError: issubclass() arg 1 must be a class
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
Discarding file:///Users/pradyunsg/Developer/pip/tests/data/src/setup_error: metadata generation failed
error: metadata-generation-failed

× Encountered error while generating package metadata.
╰─> from file:///Users/pradyunsg/Developer/pip/tests/data/src/setup_error

note: This is an issue with the package mentioned above, not pip.
hint: See above for details.
~/Developer/pip ◆ pip ·························································································· ◆ 3.10.1 better-subproce
❯ |
```

#PyConStories

...here's a stack trace that uses color **judiciously** to show exactly what you might need to do.

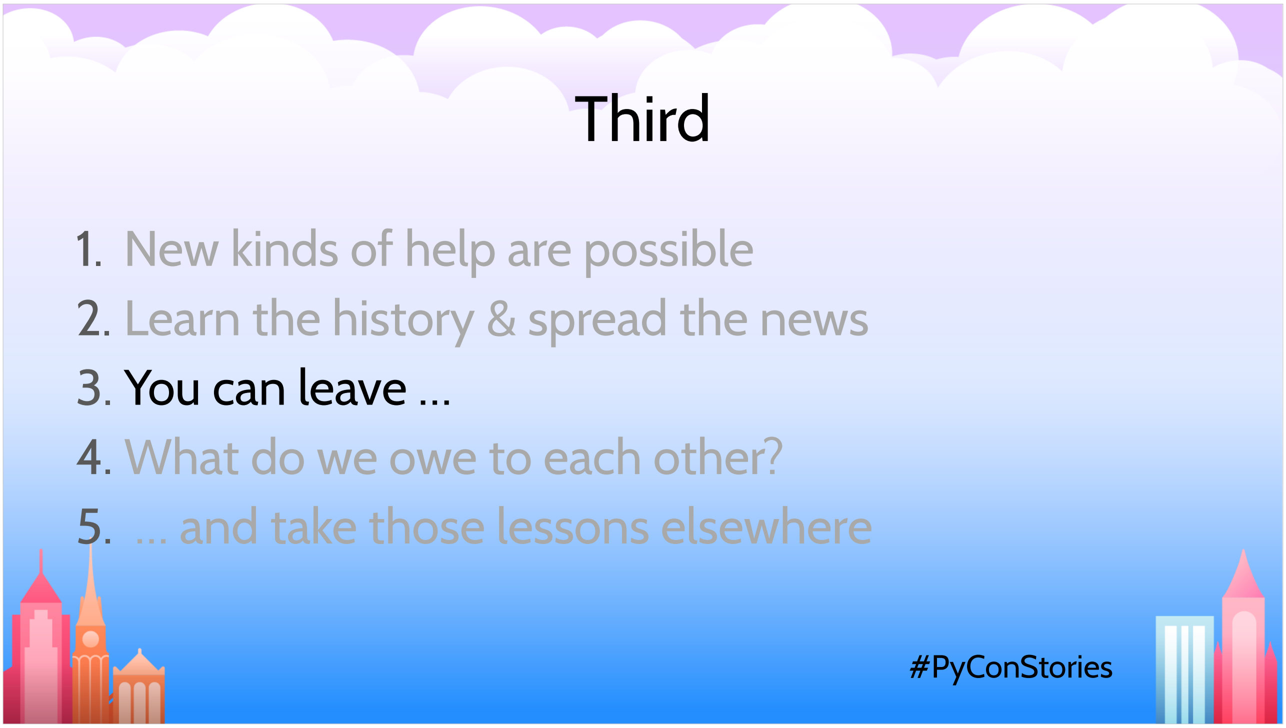


USER TESTING CAN BE FUN

A small FAQ for small open source teams!

These are some common questions we see about user testing.

Superbloom, which is the consultancy that we worked with, created user experience research case studies and guidance for us, and wrote a more general guide to how your small open source project can do user testing. (I link to that in the blog post.)



Third

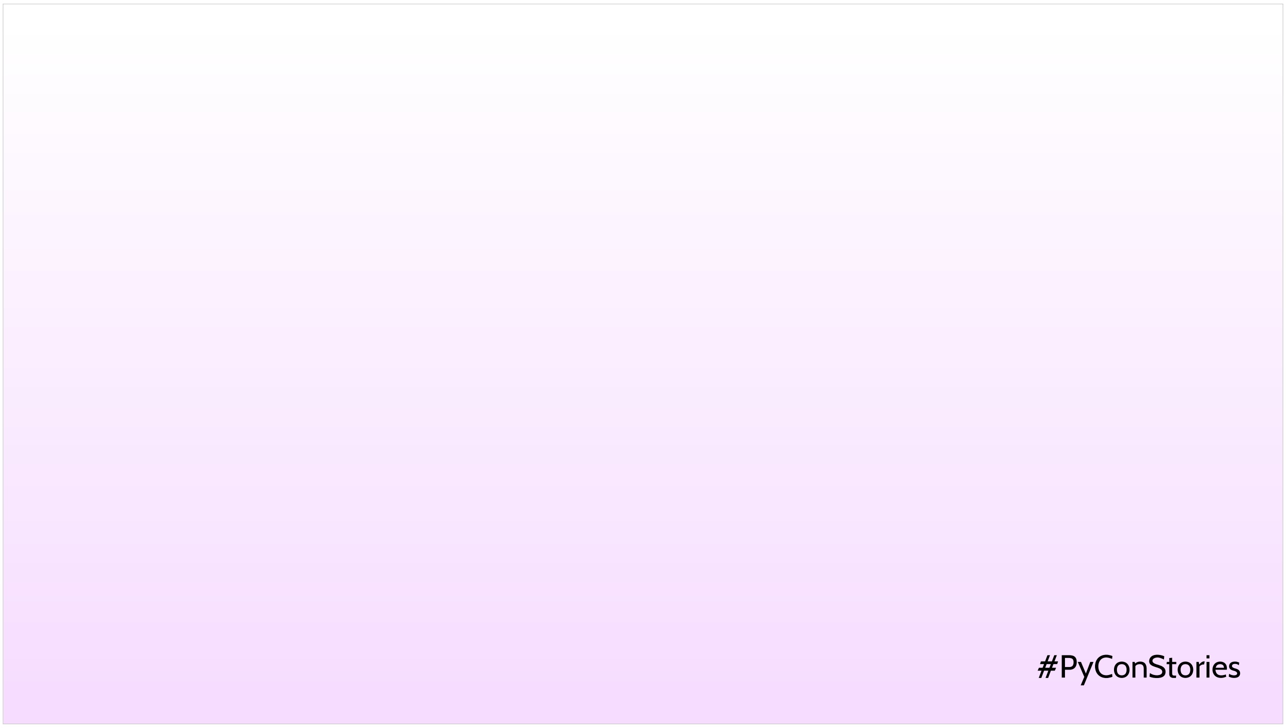
1. New kinds of help are possible
2. Learn the history & spread the news
3. You can leave …
4. What do we owe to each other?
5. … and take those lessons elsewhere

#PyConStories

I have been very encouraging, right, very PyCon, very "you can do this." But sometimes the "you can do anything" empowerment makes you feel like you have to do everything. (Maybe just me?) PyCon will have **all** these workshops on how to get into open source, but not on how to get out. [BIG LAUGH]

(There's some people here who aren't even laughing. They're just like, “yeahhhh.” [DIFFERENT BIG LAUGH])

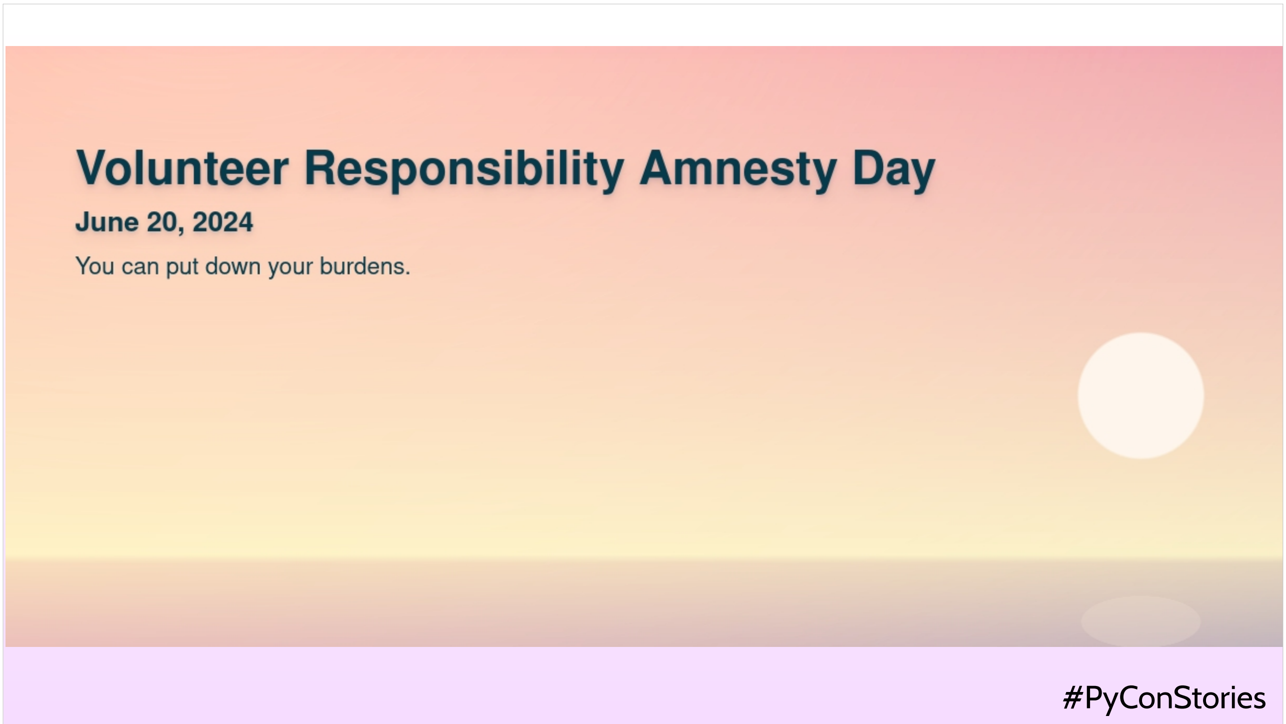
Responsibility in open source is so sticky we should make, like, epoxy out of it because you can spend six days making something and then six years trying to find someone to hand it off to. [LAUGH]



#PyConStories

I took a trip, while I was working on packaging, to co-work with a colleague for a day. The kind of person who takes on a lot of stuff. And I said, "hey, hold on. What all **do** you have responsibility for?" And we made a list on paper. And it just kept going. By the end of just **making** the list, he was like, "oh, maybe I should put a couple of things down." So he archived a repository or two, or stepped away from some moderation responsibilities on a mailing list.

People who want to make a difference often end up acting like: "just because you made a dent in the universe means you have to live there." But that is **not** the case.



Volunteer Responsibility Amnesty Day

June 20, 2024

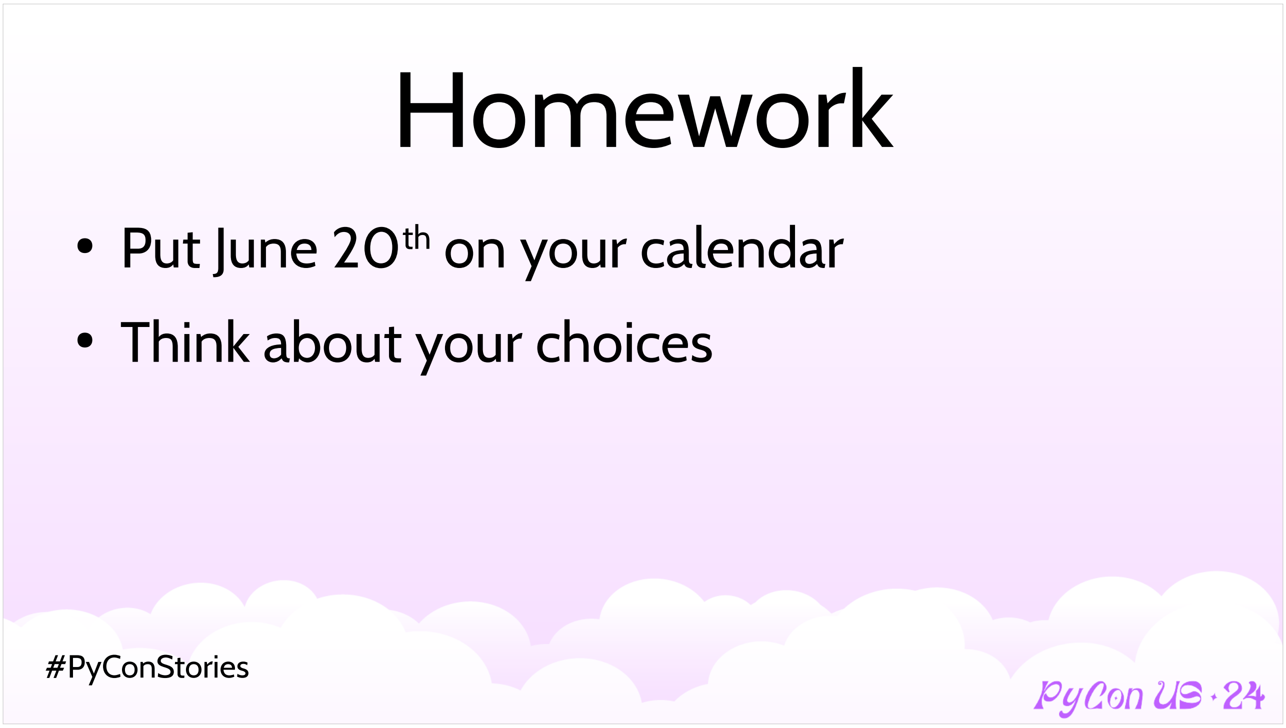
You can put down your burdens.

#PyConStories

And that is why I created Volunteer Responsibility Amnesty Day. [APPLAUSE SO BIG THAT EVEN THE WHISPER AUTO-TRANSCRIPTION CAUGHT IT]

The responsibilities that we have to our **users** are significant. And then there's the responsibilities we all have to each other, and to **ourselves**, which include, let's say twice a year, at the solstices, making a list of the stuff that you said you'd do, and then checking whether you still want to do it.

Ambitious accretion needs to be matched with thoughtful ending. There's no other way this whole thing, sorry, [PUTS A COIN OR 2 IN THE SWEAR JAR] is going to be sustainable. [LAUGHTER]



Homework

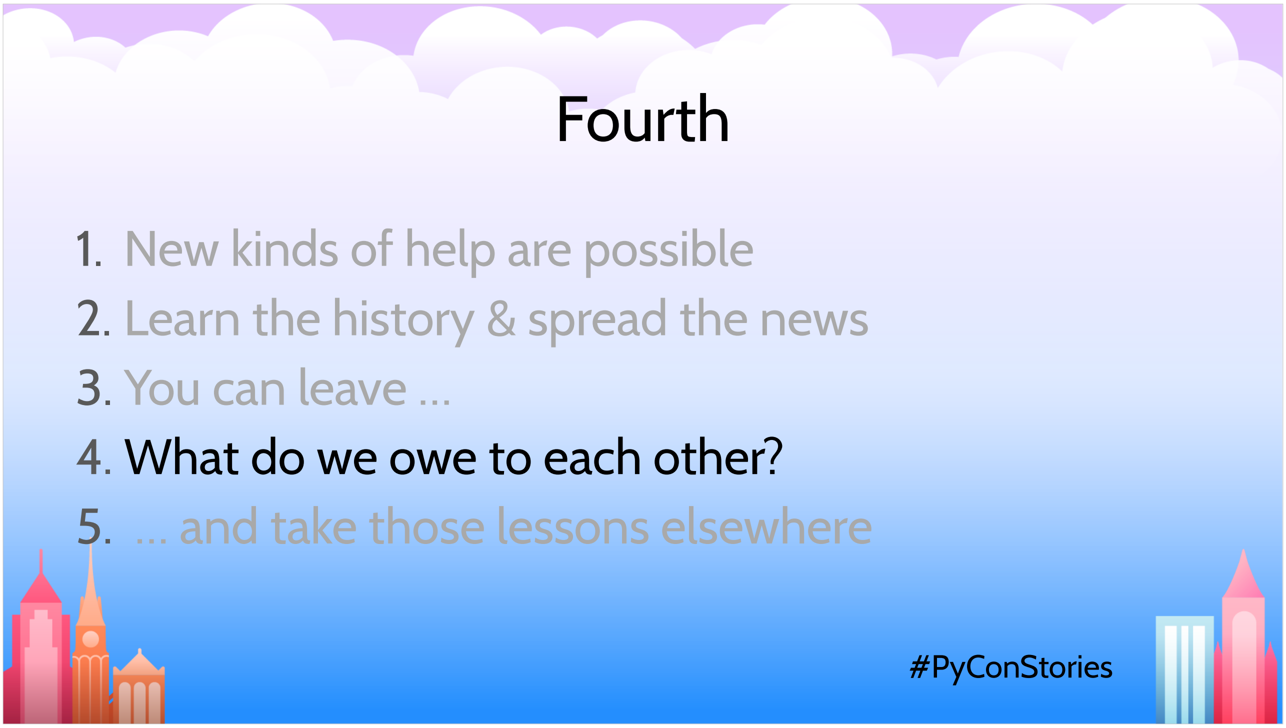
- Put June 20th on your calendar
- Think about your choices

#PyConStories

PyCon US · 24

So this is your second piece of homework.

The next of these is June 20th. Put a meeting with yourself on your calendar for 20 minutes just to think about your choices.



Fourth

1. New kinds of help are possible
2. Learn the history & spread the news
3. You can leave …
4. What do we owe to each other?
5. … and take those lessons elsewhere

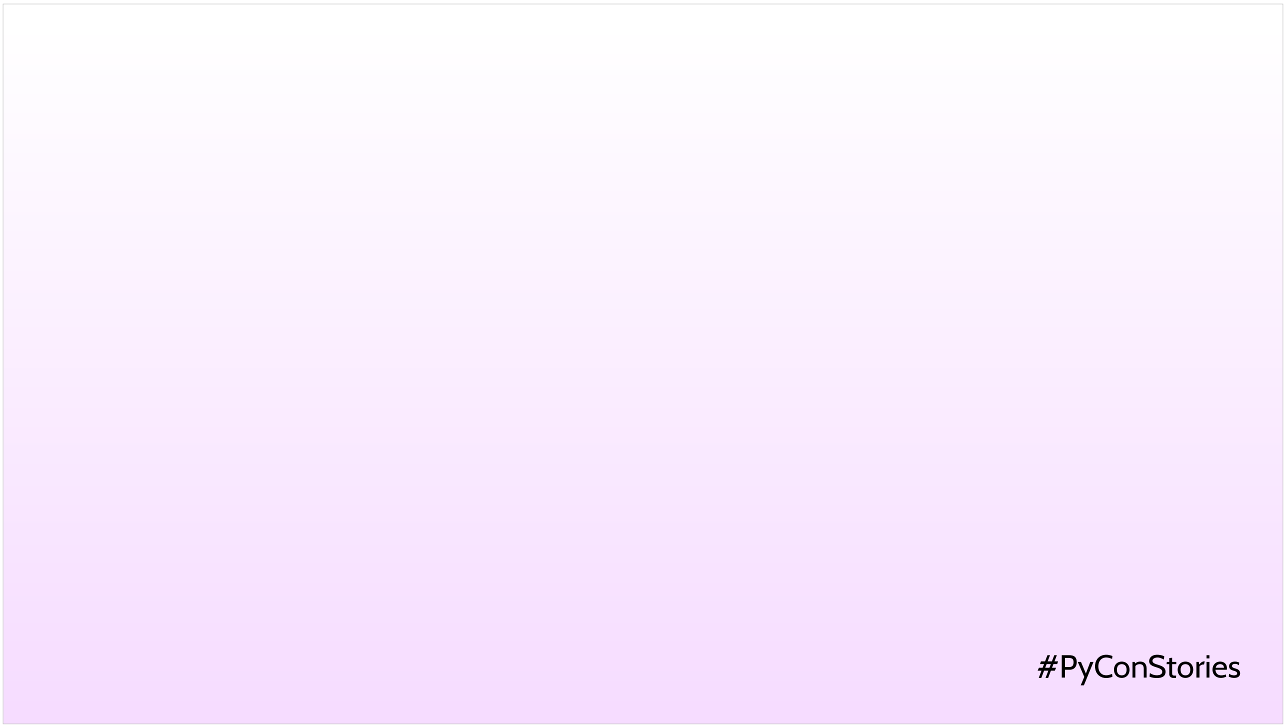
#PyConStories

If the free bit of free software means **anything**, it means that you have the freedom to **leave**. You are not obligated to carry a load on your shoulders just because you are currently, maybe, temporarily capable of doing so.

There's got to also be some **reciprocity** to this whole thing.

Those of us who were fans of a TV show called *The Good Place* recently might have a particular way that they are nodding to each other about story number four, "What do we owe to each other?" Let's think about obligation and reciprocity. Let's think about the fact that **users** are unhappy with institutions that are, that are faceless or unaccountable, where it doesn't feel like the feedback goes anywhere or is listened to. **And** it is absolutely reasonable for **maintainers** to begin to feel some imbalance, when they're giving a lot and not getting anything back.

Do we as maintainers of open source infrastructure owe the same obligation to users who give back, and to users who have the capability to reciprocate but choose not to?



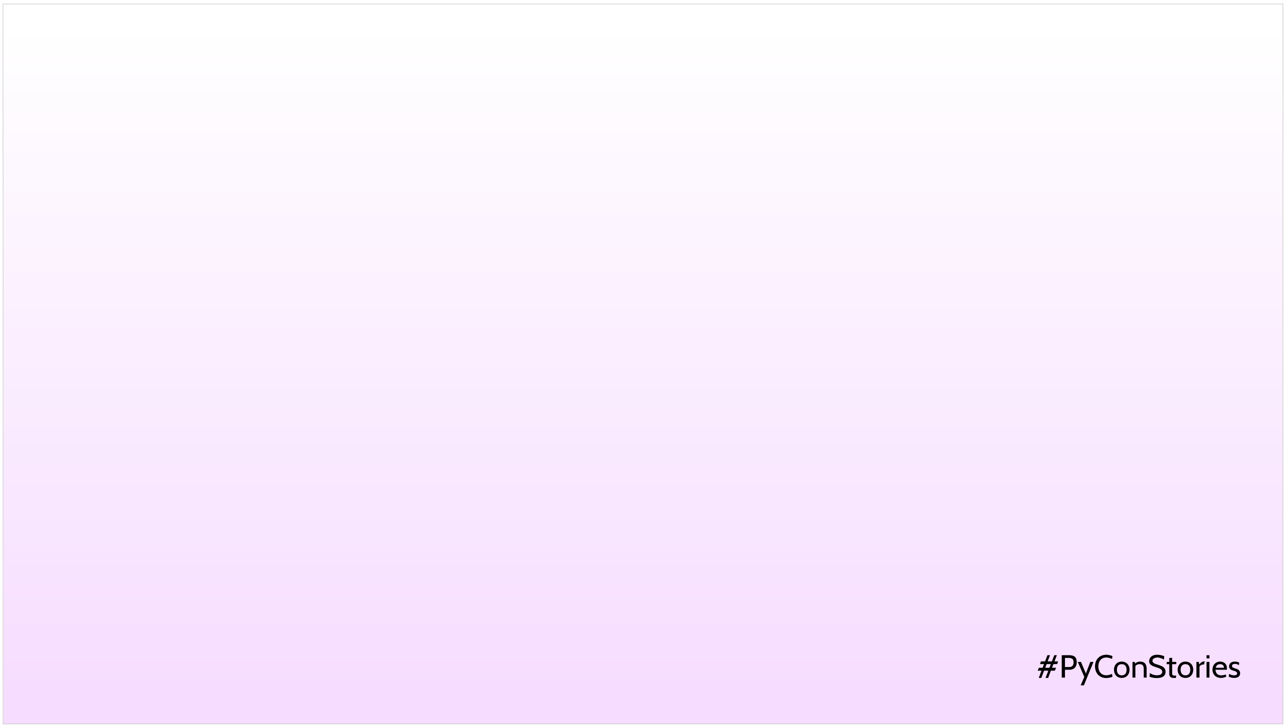
#PyConStories

We have responsibilities partly because other people don't step up and by people, I mean corporations. [PUT SEVERAL COINS IN SWEAR JAR, SIGH HEAVILY; LAUGHTER]

Uh, this is a blank slide partly so I don't name names [SIGH; PAUSE] of some companies that had real entitled things to say when we were doing various switchovers that badly needed doing, and that we were on really limited grant-funded timelines for.

Like one that -- we were doing a migration, and, you know, we had put deprecation notices in the error messages. But they weren't seeing them, because they weren't looking at the virtual machines in their continuous integration. And they were mad at **us** about that.

Or, oh hey, well, let me give you the analogy I told my mom at the time. Imagine some volunteers build a well. And, you know, it pumps some water and it's enough for some people. But it's not necessarily enough for the companies that are plugging in the pipelines for bottling plants, so that they're going to **sell** bottled water, you know, from something they got for **free**….



#PyConStories

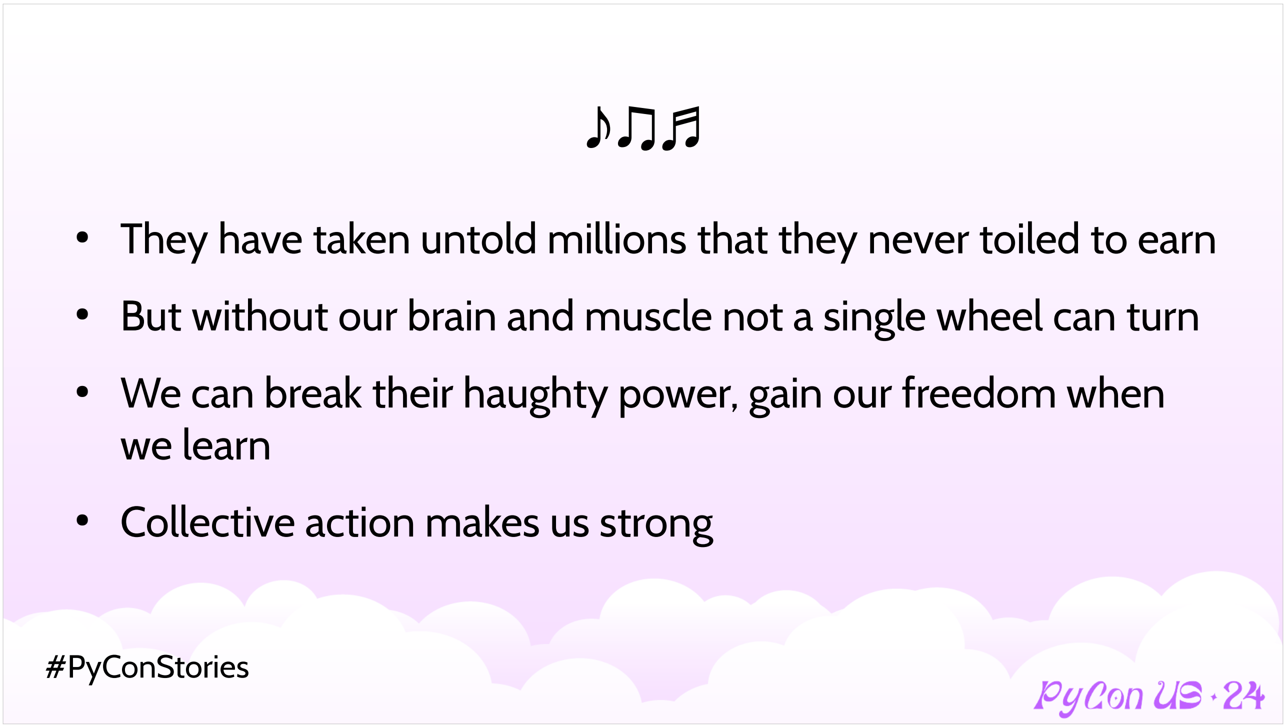
...And they aren't really paying attention when the volunteers are like, "okay, this well isn't good enough; we're not getting any money from **you** for that, but we'll go dig a better well using some non-profit, you know, grant money, a little bit that we got." And they start plastering "hey, we're going to be turning this well off on such and such a day" notices. But, you know, if you never actually physically go there, if you're **using** it but you never **go** there, you're not going to notice -- until the water starts, you know, stuttering a little bit as they're load testing, switching to the new well.

And so maybe, you know, you show up and say "What? You're switching to a new well? But my profit-making factory depends on the **old** well, currently! I certainly didn't see any of the billboards, or rather, you know, wheatpasted notices, that you put up. Surely you'll delay for, you know, months." [PAUSE]

They weren't paying attention until something happened that broke their ability to make use of the stuff we had been giving them for free, so they could make money.

Well, in the United States, we actually have a history of songs that we sing, about labor, and especially about those who make a profit by exploiting workers.

Here is one...



♪♫♬

- They have taken untold millions that they never toiled to earn
- But without our brain and muscle not a single wheel can turn
- We can break their haughty power, gain our freedom when we learn
- Collective action makes us strong

#PyConStories

PyCon US · 24

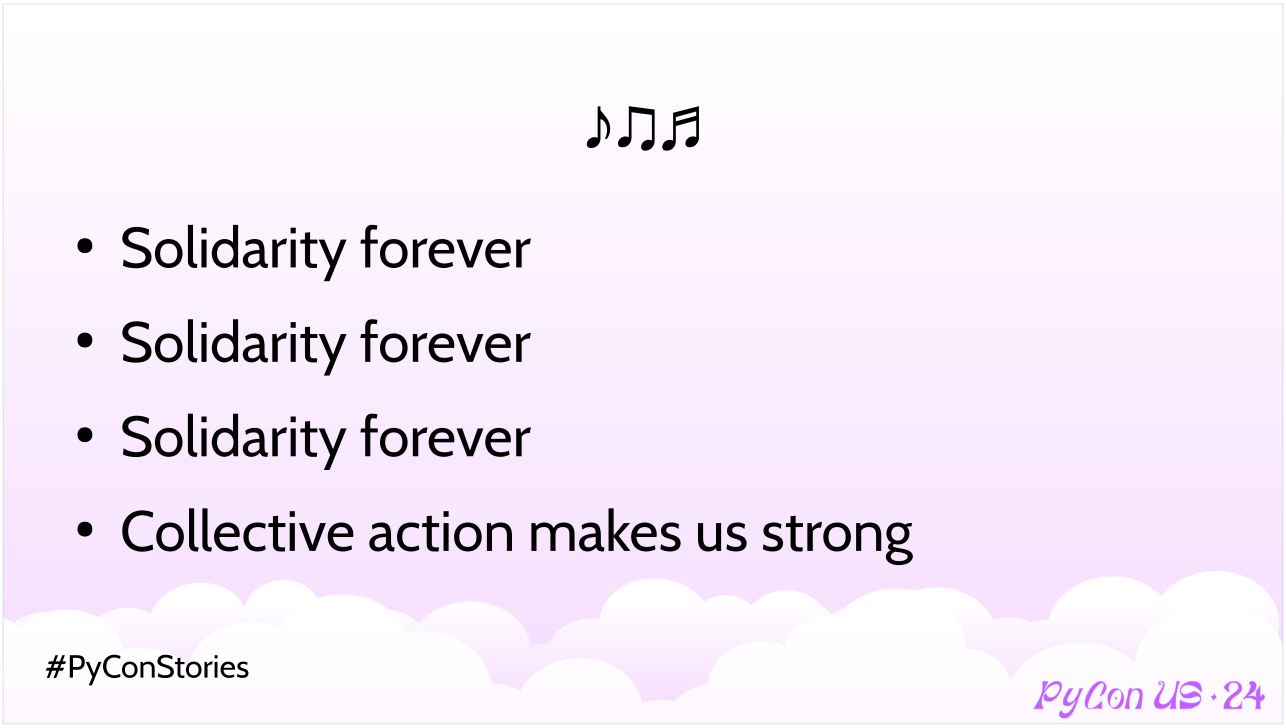
...that comes to mind for me.

[GUITARIST AND BASS SINGER ARE ONSTAGE -- I TRY THREE DIFFERENT PITCHES FOR THE FIRST WORD]

they ... they ...

[WE SING AND PLAY TOGETHER, HARMONIZING SOMEWHAT]

They have taken untold millions that they never toiled to earn
But without our brain and muscle, not a single wheel can turn
We can break their haughty power, gain our freedom when we learn
Collective action makes us strong



♪♫♬

- Solidarity forever
- Solidarity forever
- Solidarity forever
- Collective action makes us strong

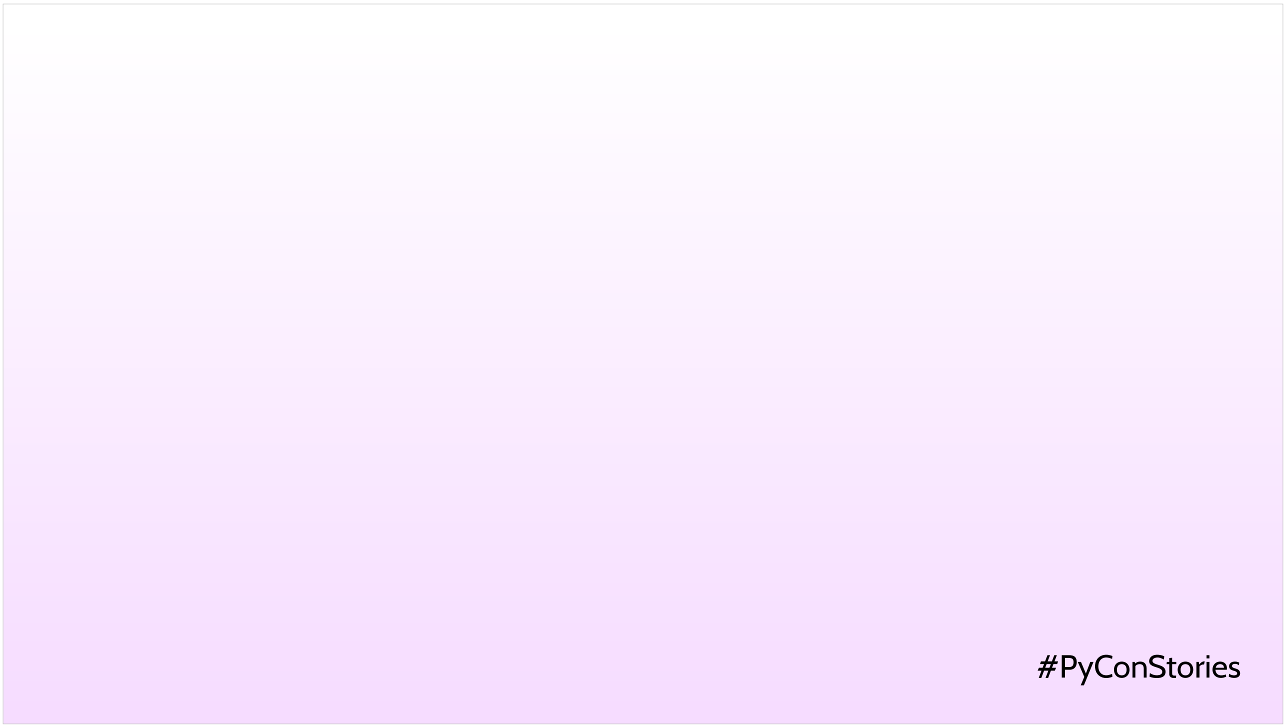
#PyConStories

PyCon US · 24

[PEOPLE IN AUDIENCE ARE CLAPPING AND SINGING ALONG]

Solidarity forever
Solidarity forever
Solidarity forever
Collective action makes us strong

[BIG APPLAUSE]

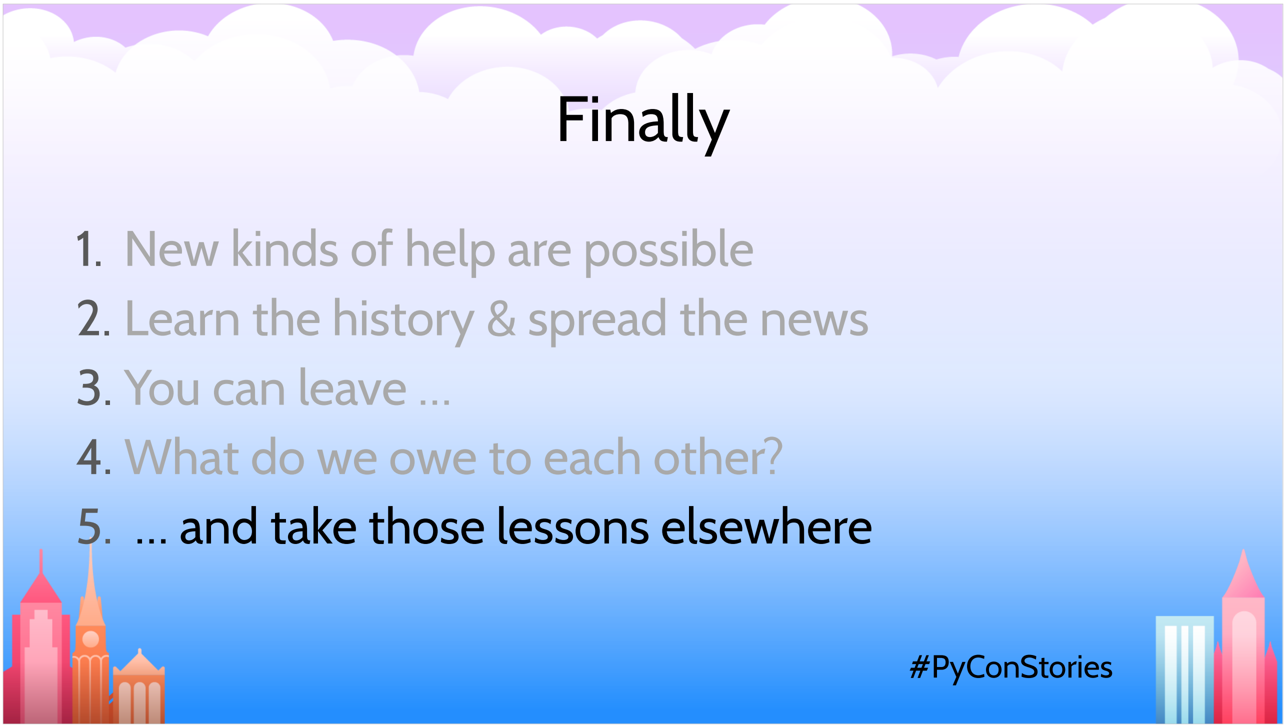


#PyConStories

Thank you Jeff Eppler for accompanying me on bass, and thank you for accompanying me on guitar to my spouse Leonard Richardson, who is also the creator and maintainer of the screen-scraping library Beautiful Soup, by the way.

[APPLAUSE]

Speaking of invisible infrastructure. [LAUGH]



Finally

1. New kinds of help are possible
2. Learn the history & spread the news
3. You can leave …
4. What do we owe to each other?
5. … and take those lessons elsewhere

#PyConStories

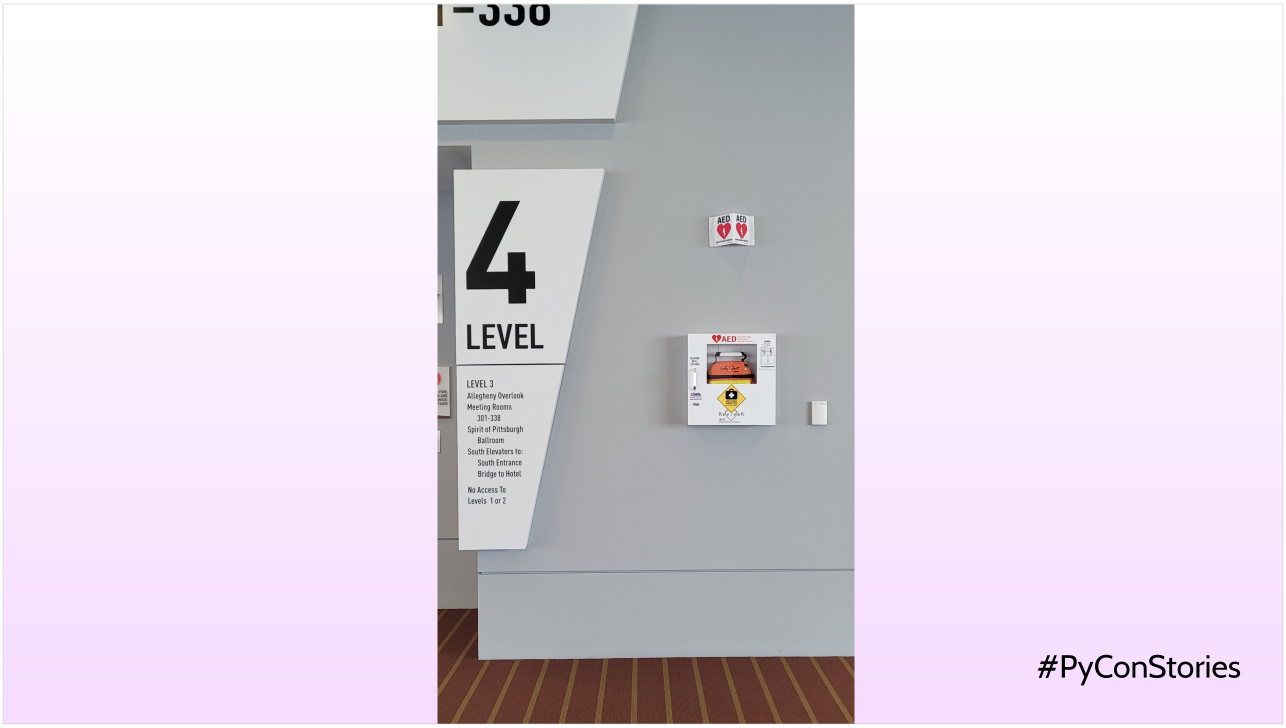
I don't work on packaging anymore. I helped get the ball rolling, but I specialize in **short-term** project management. The best thing for packaging right now is a **long-term** view. More companies sponsoring the Python Software Foundation, sponsoring PyPI, so we can pay more employees to work on the **long** term and flexibly do what needs doing for **everybody**. [BIG APPLAUSE]

But the **lessons** of working on infrastructure, I took elsewhere.

#PyConStories

Several years ago I took a CPR class (cardiopulmonary resuscitation) and I learned how amazing automated external defibrillators, or AEDs, are. If somebody has cardiac arrest, right, if they have a heart attack, CPR can do some good. But defibrillators that deliver an electrical charge can so often restart that heart. And the faster we get there with a defibrillator, the greater the chances of survival and a good health outcome.

And now, as of the last, actually, couple of decades, there are **automated** external defibrillators that can actually sense: "oh, does this person need a shock?" If you put them on a healthy person's chest, no shock delivered. But if there is an **absence** of heart rhythm, or an **arrhythmia** that could be corrected by a shock, then it will deliver that shock. How amazing that we have brought technology to restarting something that has gotten stuck, that has stopped.



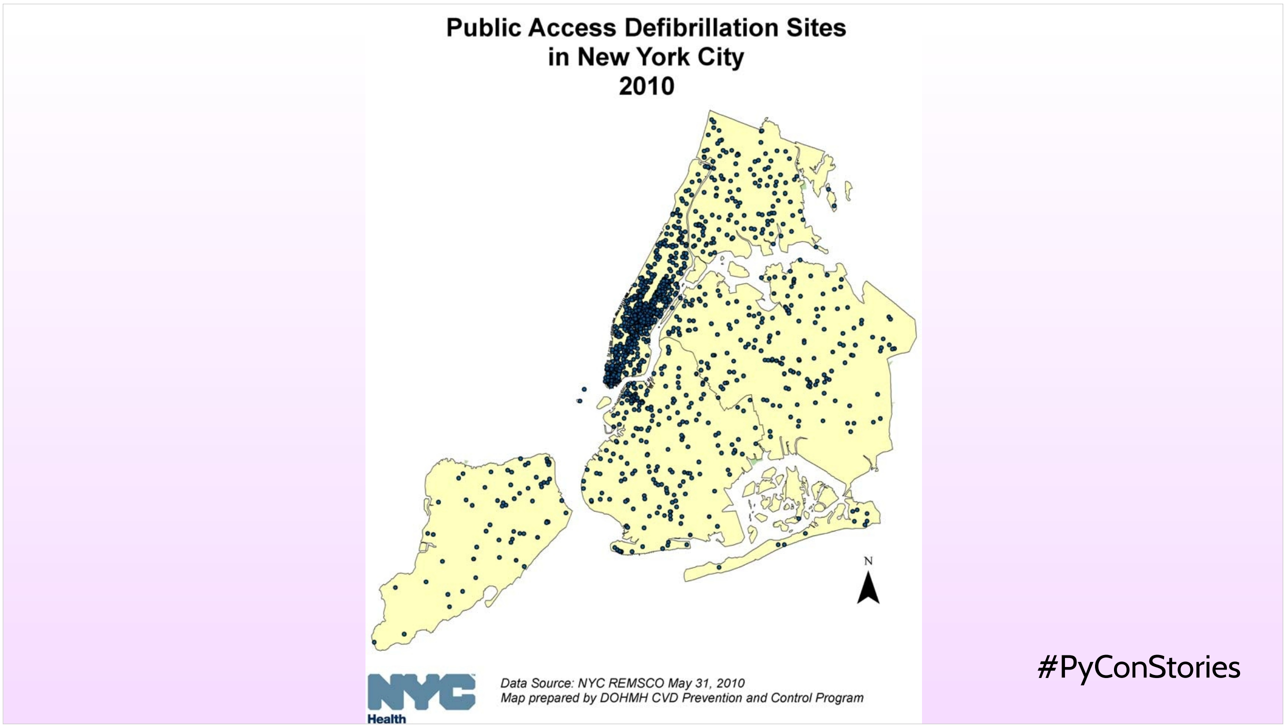
#PyConStories

And that's why they're safe enough to have on the wall at this conference center, right? And airports and gyms and train stations. Because **completely untrained people** can save lives.

Think of these not like fire **hydrants**, where you got to wait for the professionals to show up, you know, because they're the ones to use them. No, no, no. These are more like fire **extinguishers**. Anybody. **You.** If you see a problem that these could possibly address, feel free to grab it. You are not going to hurt. You may well help and help save a life, or at least keep the problem from getting worse while we're waiting for the professionals to get there.

One of these pieces of infrastructure that **is** a lot of **places**, but sometimes can **feel** invisible, because maybe our eyes just pass over them. We assume it's for someone else.

No. That's for you.



Public Access Defibrillation Sites
in New York City
2010

Data Source: NYC REMSCO May 31, 2010
Map prepared by DOHMH CVD Prevention and Control Program

#PyConStories

So I found out about these, and I thought: "gosh, that's really cool that there's so many of these. And there's a lot of **public access** ones as well in government buildings and fast food restaurants." All sorts of stuff. So I thought, "Well, where's my nearest one? I want to know: where's my nearest AED, the places I spend the most time?" Because that way, in case I see something happen, while one person does CPR, another calls 911, someone else can go run a couple minutes and grab that AED and use it.

And in a very walkable city, like New York City, where I live, sometimes that might beat the ambulance by minutes. And those minutes could save a life.

But I couldn't find out where my nearest one was. This was the best, most recent map I could find. It was very, very, coarse-grained. It did not have any addresses, or latitudes and longitudes. And it was from 2010.

And this bothered me.

So I guess I decided to scratch my own itch...

THE COUNCIL
03/30/2023
Committee on Health
constituent Sumana Harihareswara, who
has brought this issue to our attention

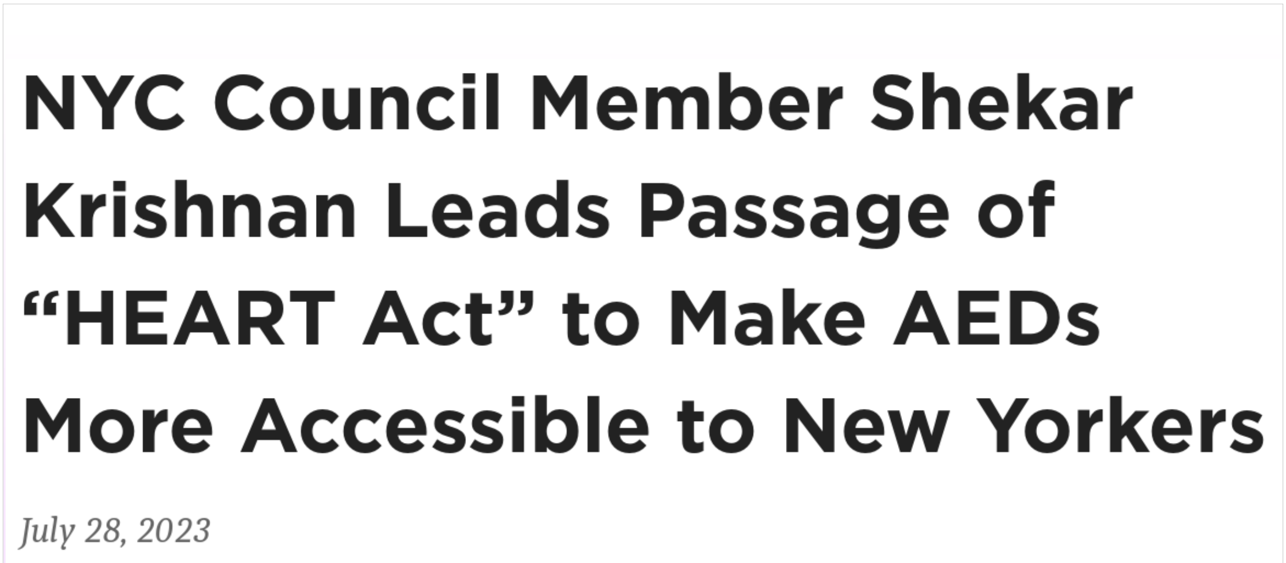
#PyConStories

... because I suggested a bill to the New York City Council.

And it wasn't workable, and so I suggested **another** bill. And then that draft wasn't actually going to achieve what we wanted, so I worked with them to revise it -- basically, I made a pull request. [LAUGHTER]

And I shepherded it along, and I worked with the maintainers.

And the way I talked about it, other people said, "oh gosh, you're pretty **passionate** about this, aren't you." And, I mean, to me, it didn't **feel** like passion. It's just like, "well, **this** just isn't right. And something's got to **happen** about it, right?"



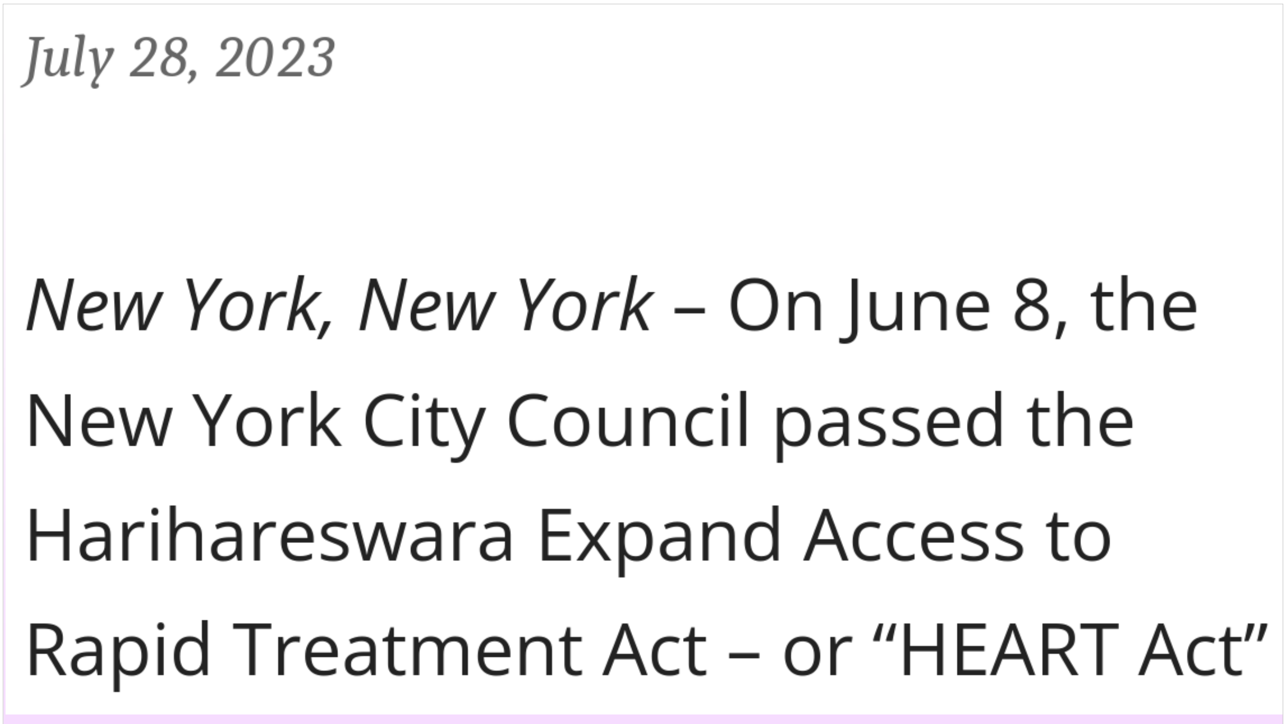
NYC Council Member Shekar Krishnan Leads Passage of “HEART Act” to Make AEDs More Accessible to New Yorkers

July 28, 2023

#PyConStories

So we revised the bill and the city council passed it. [APPLAUSE]

It's the HEART Act...



July 28, 2023

New York, New York – On June 8, the New York City Council passed the Harihareswara Expand Access to Rapid Treatment Act – or “HEART Act”

...the Harihareswara Expand Access to Rapid Treatment Act. [APPLAUSE] It's named after me, and it's named after my father, who died of a heart attack.

They were **supposed** to release this data on January 1st, 2024, and they have not done so yet.

It's a big complicated system, and it's not working right.

And I've learned before that even someone like me, who didn't know that domain going in, can make a difference.

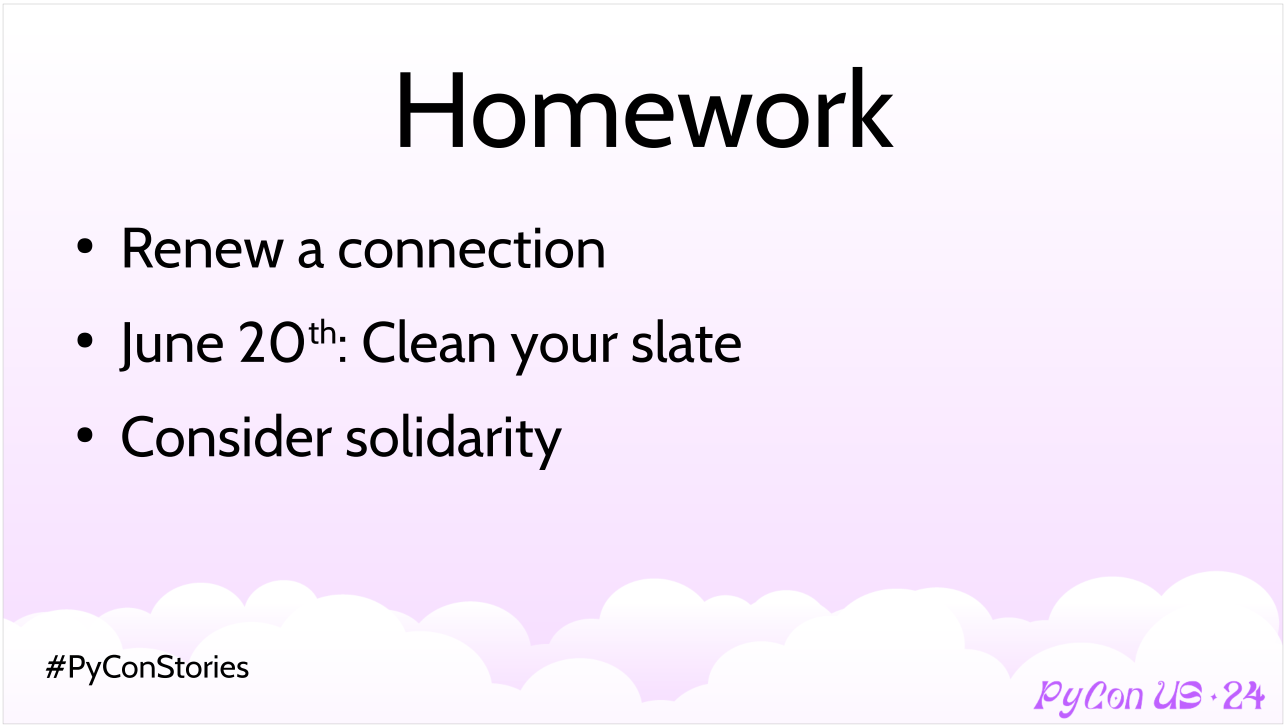
I'm not a public health expert. I'm not a data analysis expert. I'm not an emergency medicine expert. But I wasn't a Python packaging expert, either, right?

And I learned, when I worked on infrastructure: there are big complicated systems that emerged over time. And sometimes there's a part of them that doesn't work just right. And I can see how I can make a difference. By looking around, thinking about what our goals are, understanding what activities we're doing towards those goals, who's in charge of what, what resources are available, and finding the right question to ask, and not going away until we get a good answer.

Because I am curious and I am stubborn. Because I am a project manager.
[APPLAUSE]

There are infrastructures in your world, whether invisible **digital** infrastructure or something as obvious as a big box on a wall, that you could engage with. You could engage with something that seems so big. And you could be a defibrillator. You could be those paddles that revive something that's not working so great. Whether in open source or local government or something else.

You can awaken your curiosity and your stubbornness. Find the right place to engage with something that seems big. Something infrastructural, something other people don't even realize they depend on. And you could make a difference maybe no one else even realizes needs to be made.



Homework

- Renew a connection
- June 20th: Clean your slate
- Consider solidarity

#PyConStories

PyCon US · 24

So I have three pieces of homework for you.

One: renew a connection with someone you miss. Life is short.

Two: a month from now, June 20th, clean your slate. Look through your volunteer responsibilities. Archive or reduce or step away from the things that just aren't working for you anymore. Let's make continuing to do these things as intentional as starting them was. Or more so.

And so that:

Three: you can have some more room to consider solidarity. We are all in this together. [APPLAUSE]

Infrastructure is how we care for each other at scale. And I have been so privileged to be able to take care of so many of you. And I hope you can join me.

Thank you

- Anonymous
- Audrey
- Brendan
- Candra K. Gill
- Colin
- David Renshaw
- Dustin Ingram
- Ee Durbin
- Elizabeth Yalkut
- Federico Mena Quintero
- Heidi R Waterhouse
- Jacob Kaplan-Moss
- James S
- Jed Hartman
- Jeff Eppler
- Jing
- julia evans
- Leonard Richardson
- Lilly “attacus” Ryan
- Nick Murphy
- Pavithra Eswaramoorthy
- Pradyun Gedam
- Zed Lopez

PyCon US · 24
By the community, for the community

I want to thank the captioner, the interpreters, all the people who gave feedback on the talk, Leonard and Jeff for accompanying me...

Thank you

PyCon US · 24

By the community, for the community

...thank you, and thank you PyCon.

[STANDING OVATION]

[I TAKE A BOW WITH LEONARD AND JEFF]

[CONFERENCE CHAIR MARIATTA, RETURNING TO STAGE]:

All right. Thank you so much, Sumana. I have my calendar here. I am going to put it on my calendar. June 20. There you go.