

How can we have healthier technical discussions?

Nathaniel J. Smith / @vorpalsmith
PyCon 2019 Maintainers Summit

Topics covered in open-source development guides →

Choosing a license

Coding

Testing

Debugging

Code review

Version control

Packaging

Supporting users

Triaging issues

Documentation

Onboarding newcomers

Marketing your project


Governance

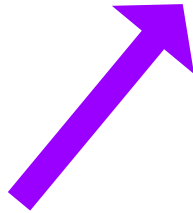
Codes of conduct

**What
open-source
developers
mostly do →**

Argue on
mailing lists

Decide on what the resolution rules for `__op__`/`__rop__`/`__numpy_ufunc__` actually are #5844

 Closed njsmith opened this issue on May 6, 2015. **324 comments**



Distutils-SIG - July 2017 Archives by thread

- Messages sorted by: [\[subject \]](#) [\[author \]](#) [\[date \]](#)
- [More info on this list...](#)

Starting: Sat Jul 1 06:53:34 EDT 2017

Ending: Sun Jul 30 09:12:49 EDT 2017

- [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nathaniel Smith*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Paul Moore*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nick Coghlan*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Daniel Holth*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Donald Stufft*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nathaniel Smith*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nick Coghlan*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Thomas Kluyver*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Paul Moore*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nick Coghlan*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Thomas Kluyver*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nick Coghlan*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Donald Stufft*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Thomas Kluyver*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Donald Stufft*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nathaniel Smith*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nick Coghlan*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Donald Stufft*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nick Coghlan*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Donald Stufft*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Donald Stufft*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Paul Moore*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Thomas Kluyver*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Paul Moore*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nathaniel Smith*
 - [\[Distutils\] A possible refactor/streamlining of PEP 517](#) *Nick Coghlan*



Notice
scrollbar

Should Python support := as an inline assignment operator?

- Chris .. Nick .. (23) [Python-Dev] The new and improved PEP 572, same great taste with 75% less complexity
- Łukasz .. Steve .. (63) [Python-Dev] (name := expression) doesn't fit the narrative of PEP 20 - On Thu, Apr 26, 20:
- Ryan .. Stephen .. (10) [Python-Dev] Is PEP 572 really the most effective way to solve the problems it's target... -
- Fatty, Lukasz, Steven (3) [Python-Dev] Reserve ':= ' for type-inferred variable initialization (was PEP 572) - Hi Fatty, a
- Martin .. Terry, Greg (6) [Python-Dev] PEP 572: Why not := as standard assignment operator? - Hi list, when readin
- Jim J. Jewett [Python-Dev] Nickname Binding (PEP 572) - I think part of the disconnect is that this enhancer
- Sven .. Sven, Chris (10) Re: [Python-Dev] PEP 572: Assignment Expressions - On 25.04.2018 01:19, Steven D'Aprar
- Victor .. Stephen .. (22) [Python-Dev] PEP 572: Write vs Read, Understand and Control Flow - Chris Angelico writes
- Victor .. Antoine, Carol (7) [python-committers] How to calm down the discussion on the PEP 572? - Hi, Since 2 or 3 y

[python-committers] Transfer of power

Guido van Rossum [guido at python.org](mailto:guido@python.org)

Thu Jul 12 10:57:35 EDT 2018



Now that PEP 572 is done, I don't ever want to have to fight so hard for a PEP and find that so many people despise my decisions.

I would like to remove myself entirely from the decision process. I'll still be there for a while as an ordinary core dev, and I'll still be available to mentor people -- possibly more available. But I'm basically giving myself a permanent vacation from being BDFL, and you all will be on your own.

Discussions are crucial

Technical side

They're how we make decisions.

Human side

They're a huge source of burnout.

Why are there 100x more resources on choosing a license, when we spend 100x as much energy on discussions?

Okay but how?



What makes a good discussion?

Technical side

Choose the best solution.

Human side

Don't make people miserable.

Pick one?

The marketplace colosseum of ideas



Two+ proposals enter, one proposal leaves!

For most software problems:

- There are thousands of potential solutions,
- that interact with thousands of other components,
- in thousands of different ways.

Consequences:

Technical side

No matter how smart you are, you probably can't find or recognize the best solution.

Human side

Our careers and self-worth are based on lies.

What makes a good discussion?

Technical side

Choose the best solution.



Integrate input from as many perspectives as possible.

Human side

Don't make people miserable.

Complicating the Narratives

What if journalists covered controversial issues differently —based on how humans actually behave when they are polarized and suspicious?



Solutions Journalism

Jun 27, 2018 · 39 min read

By Amanda Ripley [updated Jan 11, 2019]

When my son rages that he has to go to bed, I now say something extremely unremarkable like, “You’re really frustrated that you can’t stay up.” Before, I would have ignored him or argued with him, explaining how reasonable his bedtime is and how lucky he is to have a bed, etc. Now, after I show him that I’ve heard him, he just...goes to bed.

What makes a good discussion?

Technical side

Choose the best solution.



Integrate input from as many perspectives as possible.

Human side

Don't make people miserable.



Make sure everyone feels heard.

What makes a good discussion?

Technical side

Choose the best solution.

Human side

Don't make people miserable.

Key metric for good discussions:
how many people feel heard?



Why aren't people heard?

- They don't know the discussion is happening
- They know, but feel unwelcome in the room
- They feel welcome, but don't speak up because they worry they don't have anything to contribute
- They speak up, but get brushed off
- They speak up, and someone listens, but only to dismiss them
- They speak up and are heard, but they can't *tell* they're being heard, so they stop talking
- So many people speak up at once that none of them can be heard

Who else knows about this?

- Mediators
- Negotiation theory
- The Quakers
 - See also: Sue Gardner's articles on "What Wikimedia Can Learn From the Quakers"
([part1](#), [part2](#))
- ... probably a bunch more I've never heard of?

**Technical
discussions are a
foundational
programming skill,
like testing or
version control.**