An Acknowledgment of Place

mertz@gnosis.cx
gnosis.cx/publish

I am humbled to be invited to this place where we are speaking together.

I know that this city where I am talking to you became a city, created by the Awori Yoruba people, in what we call the 15th century of the Western and Christian calendar.

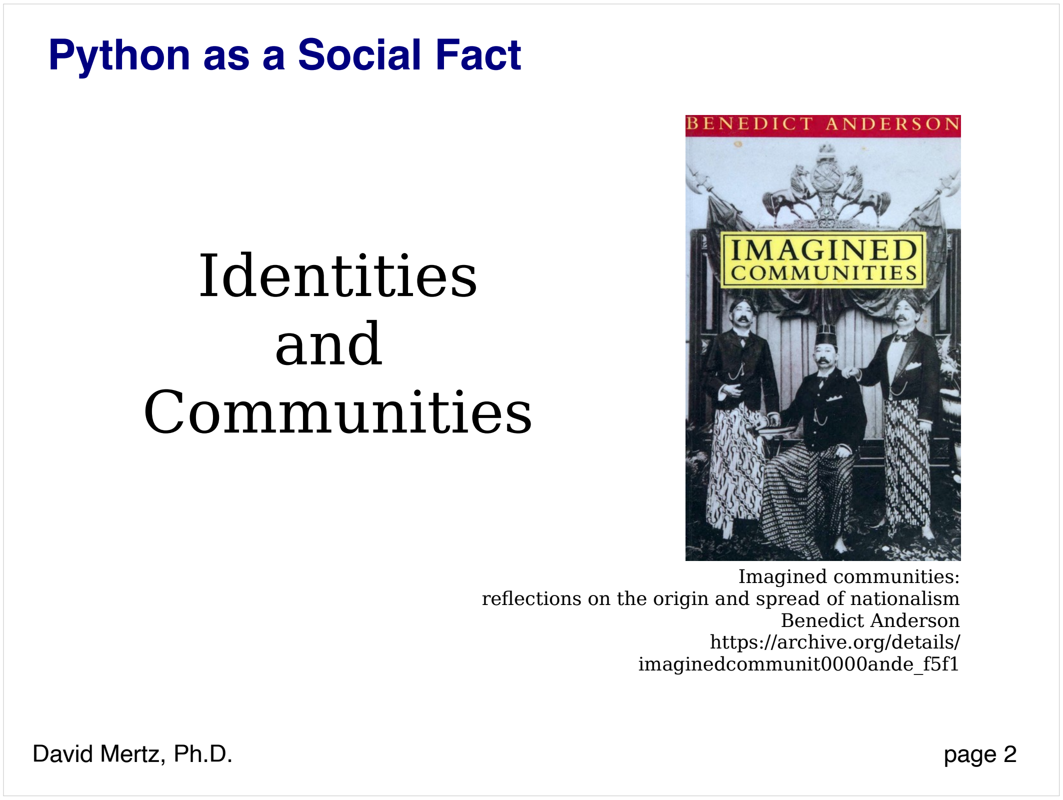
I know that this place was called Eko, under the administrative domain of the Kingdom of Benin, from the 16th century.

I know also that it was later branded as Onim, and later Lagos, by Portuguese merchants, whose successors were war criminals of the transatlantic slave trade, beginning in the 17th century.

I know that at the start of the 19th century the British Empire sought, by means of violence, both to end the enslavement of human being in Lagos, and elsewhere, and to impose imperialist control of resources and human beings under a Capitalist system rather than as *de jure* enslavement.

I know that by the time the now nation of Nigeria gained a degree of self rule in 1954 and independence in October 1960, Lagos was a diverse city across ethnic, religious, linguistic, gender, class, and other dimensions.

And of course I know that this place is home to many friends and brilliant colleagues, present and future, some of whom write Python code.



Python as a Social Fact

Identities and Communities

Imagined communities:
reflections on the origin and spread of nationalism
Benedict Anderson
https://archive.org/details/imaginedcommunit0000ande_f5f1

The tradition of “land acknowledgment” is common among progressive speakers in areas where colonialism has displaced, and often murdered, indigenous peoples.

Speaking here is not the same as speaking in the United States where I live, and *place* cannot have entirely the same insinuations. What it means to belong to Lagos differs from what it means to belong to Los Angeles.

In acknowledging my place, I have necessarily cast a net over many *identities*. We each belong to many communities, some voluntarily and volitionally, others that feel inevitable and immutable.

Benedict Anderson’s lovely 1991 book has meant a great deal to me. While I won’t encourage copyright violation, I do wonder if a PDF of his 2006 reprinted 2nd edition exists online.[1]

In this admittedly dense text, Anderson looked at how, by imagining nations, their subjects or citizens became able to imagine themselves as reflections of such national communities.

Many identities work in a similar way, and the the communities with which we identify become something more than simply an enumeration of the people belonging to them.

[1] https://is.muni.cz/el/1423/podzim2013/SOC571E/um/Anderson_B_-_Imagined_Communities.pdf

The Python Community Before There Was a Python Community

- Used Python from 1998
- Wrote about Python since 1999
- PSF Fellow from 2008
- PSF Director from 2009-2015
- Erstwhile Chair Outreach & Education Committee

These bullets perhaps look like interview boasts, but that's not my goal. I've already been invited onto this stage.

There are people in this room who are more important than I am to the Python *language*, and indeed who are more important for the Python *community*. There are still more of you who will *become* so.

However, I've been in this community, and close enough to it, for long enough to see the Python community become an identity that feels close to the heart of hundreds of thousands of people in the world, perhaps to millions.

In 1998, and even in 2008, Python was simply a useful technical tool, as it remains exclusively such for many people. However, somewhere around that latter date, it became also something people *belonged to*. A community.

What changed? And what does it mean for a community to come into existence, generally?

A Group before a Community Forms

https://pycon.blogspot.com/
2017/04/
python-1994-recollections
-from-first.html

The earliest meeting of Python developers was in November 1994. This low resolution photograph is the best I could find. I can make out several people I know, and a few others I can guess about.

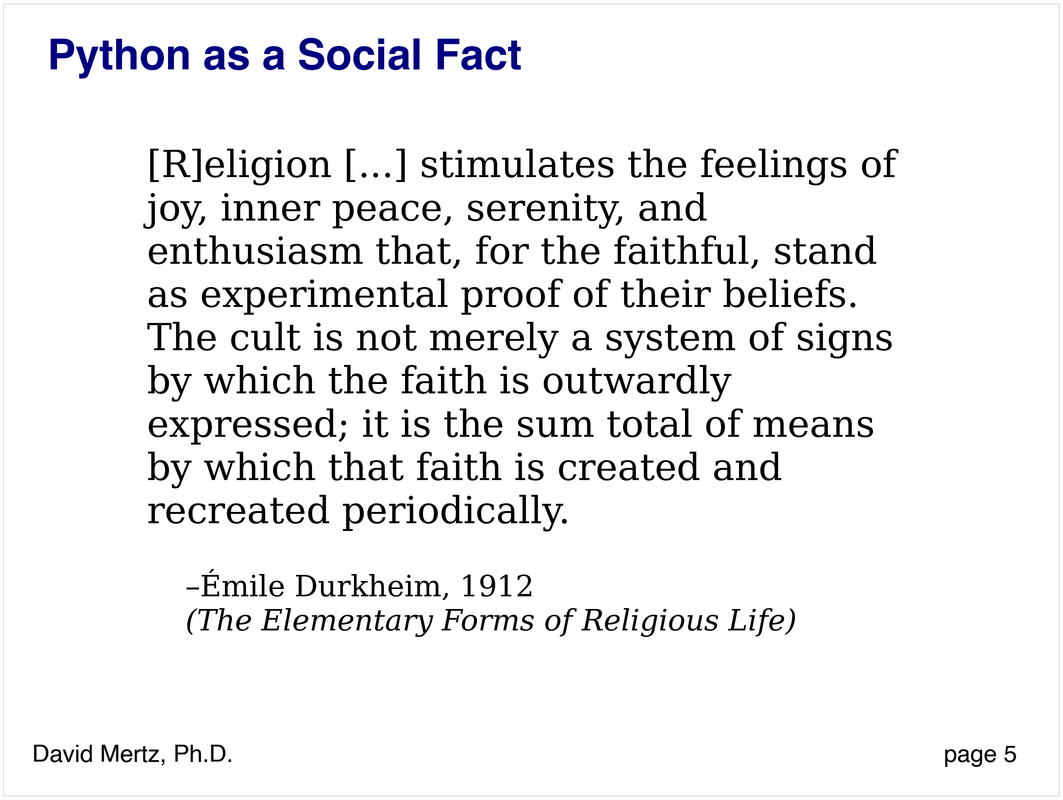
30 years ago—before I had myself heard of Python—I would describe Python as an idea, perhaps an incipient doctrine.

The Zen of Python was yet to be written. Discussions about the programming language occurred on the Usenet, but were predominantly about technical design questions, and rarely, if ever, about community or identity. Only uncommonly were they even about the "philosophy" of the Python language.

At a glance, even in low resolution, we can also see that these nine people were able-bodied 30-something white men. Most are Americans, with 2 or 3 Europeans. All were highly educated and economically comfortable.

I do not wish even remotely to suggest that early Pythonistas were messianic, or that our Python community is itself a religion.

Nonetheless, at about the same age as these men, Siddhartha, and Jesus, and Mohammed, peace be upon him, each had a small number of associates who were demographically similar to each other, and who were motivated by an idea, before their religious communities emerged. This analogy can be productive.



Python as a Social Fact

> [R]eligion [...] stimulates the feelings of joy, inner peace, serenity, and enthusiasm that, for the faithful, stand as experimental proof of their beliefs. The cult is not merely a system of signs by which the faith is outwardly expressed; it is the sum total of means by which that faith is created and recreated periodically.
>
> –Émile Durkheim, 1912
> *(The Elementary Forms of Religious Life)*

I want to lean into this metaphor of religious identity just slightly more. *Of course* I recognize that our Python community, and other programming language communities, are also very different from religious identities in many concrete and profound ways.

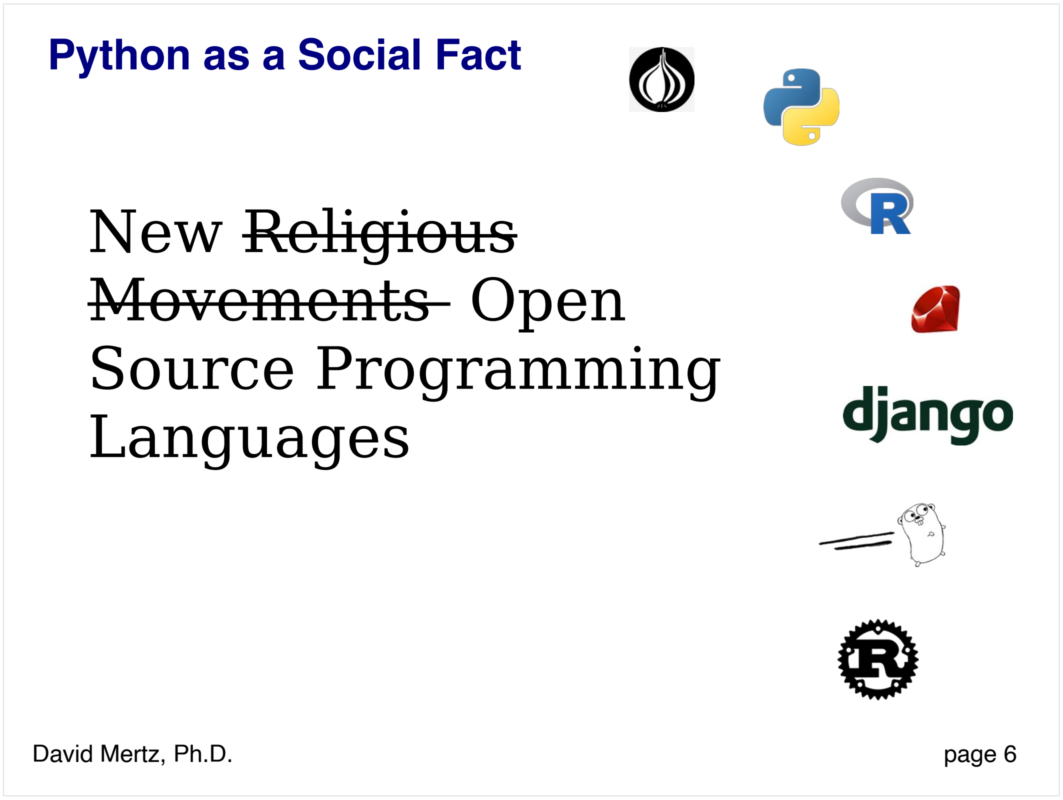
For me—and for many others—the Python community has become a source of joy, inner peace, serenity, and enthusiasm. It is something I have come to identify with deeply. And something that I continually recreate by actions such as giving talks like this.

In a small way, everyone participating in this conference is creating and recreating communities, around Python, around region and nation, around open source, and in a complex web of affinities and identifications.

My identification is because of my biographical journey, in part. However, my identification is also because of what Python has become as a social entity, something with qualities and values that conform with my own; but also that simultaneously shape my own.

A similar positive entity has developed among the communities, and formal organizations, attached to other programming languages. I think Python has often led the way in these social aspects; and many communities that are not those surrounding Python lack some of the beliefs and practices I value.

I imagine that I am much like the rest of the people in this room in my values. To be clear, in saying “I imagine” I am not simply rhetorically introducing a fact, but rather describing my relationship to this imagined community.



Python as a Social Fact

New ~~Religious Movements~~ Open Source Programming Languages

David Mertz, Ph.D.

In the 19th and 20th centuries a variety of new religions were created.

In the United States, the Latter Day Saints emerged in 1830, the Unity Church in 1891, and Scientology in 1954. In Iran, Bábism began in 1844, evolving into the Bahá'í Faith in 1863. In Korea, we saw Donghak in 1860, and its 20th century offshoot Cheondoism. This list is not exhaustive, of course.

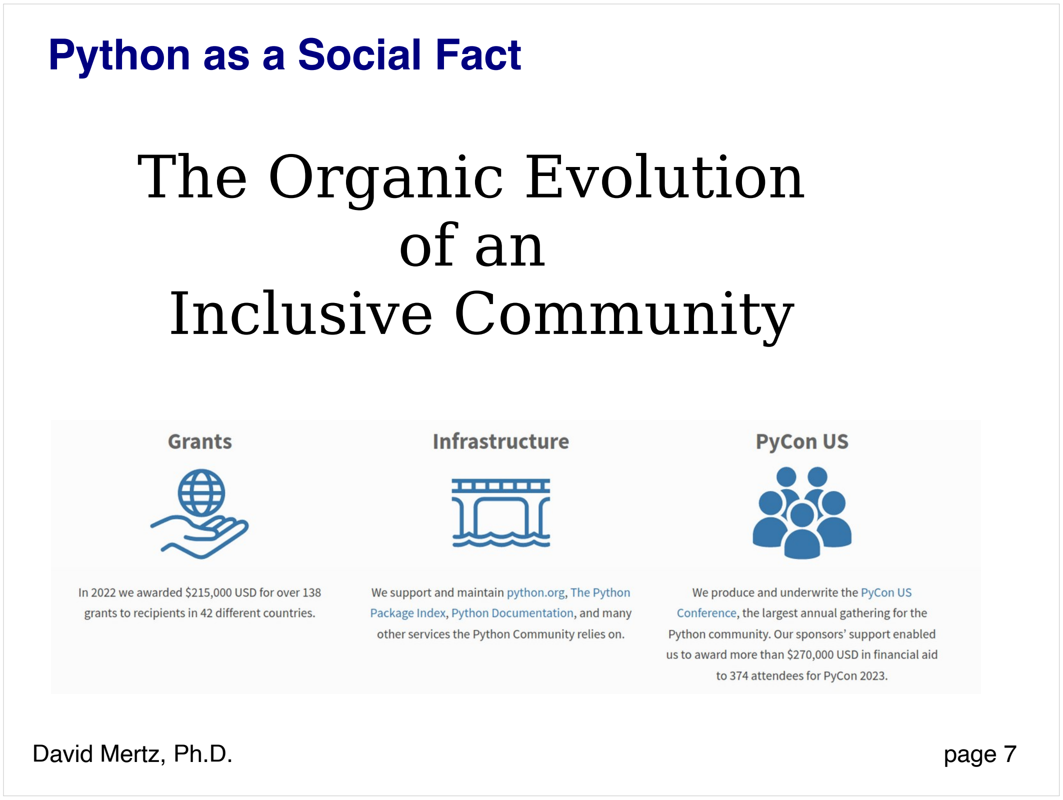
In social dynamic, during their very first years, these New Religious Movements often resembled the major world religions mentioned a few moments ago; in theology and doctrine, of course, they have differed significantly from earlier traditions.

It sometimes happens that someone identifies with the theological tenets of a new religion, and joins for that reason. But I believe it is far more common that membership follows our social perception of the communities and social networks we join.

Over the last 35 years or so, since Free Software and Open Source have become articulated and compelling doctrines, many programming languages have been created, many of which could have early photographs akin to that I showed of the 30-something aged Python creators from 1994.

Free Software has articulated a good collection of tenets. Not always perfect, and commonly enough with distinctly toxic threads as well, but far better than its opposites. While the ideas of Free Software have often been resisted by proprietary technology companies—the Microsofts, and Oracles, and Apples, and Teslas, and SAPs, and Adobes of the world—these good ideas have generally won the world.

Nearly nothing that runs on computers does so without using somewhere between a lot and entirely Free Software.



Python as a Social Fact

The Organic Evolution of an Inclusive Community

Grants

In 2022 we awarded $215,000 USD for over 138 grants to recipients in 42 different countries.

Infrastructure

We support and maintain python.org, The Python Package Index, Python Documentation, and many other services the Python Community relies on.

PyCon US

We produce and underwrite the PyCon US Conference, the largest annual gathering for the Python community. Our sponsors' support enabled us to award more than $270,000 USD in financial aid to 374 attendees for PyCon 2023.

I do not want to disparage other technical communities. It would both be outside of my direct experience and show an unnecessary lack of respect.

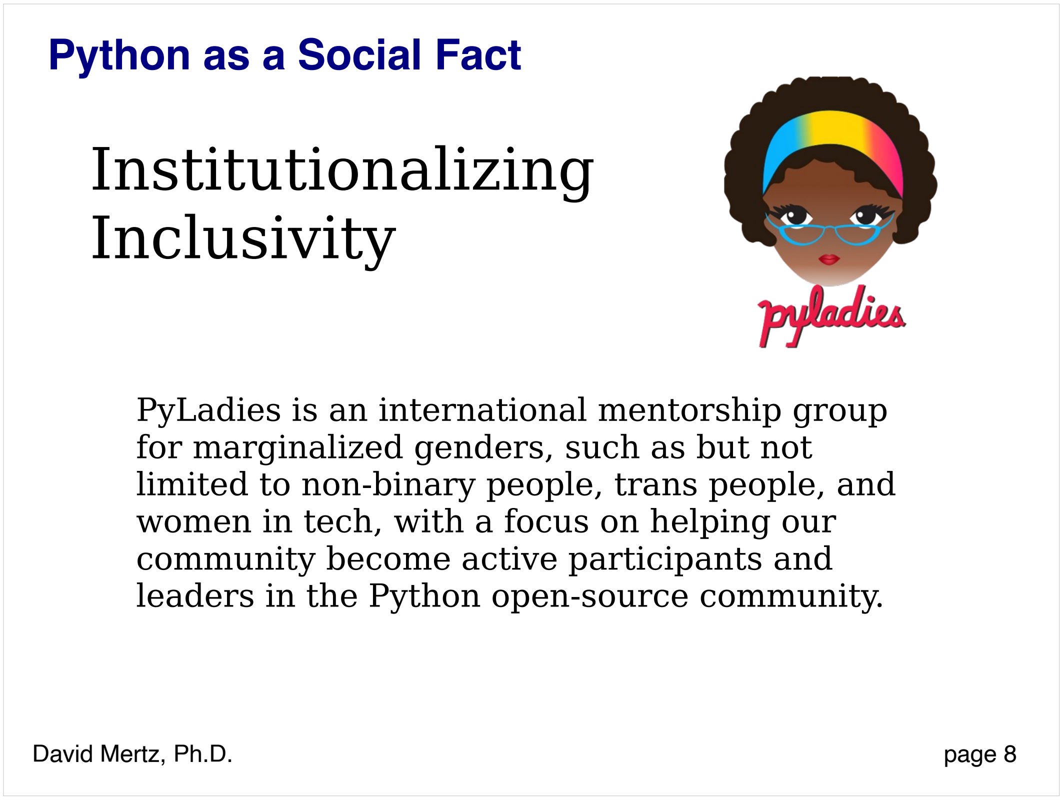
That said, in my experience among varying technical communities, there are things that make some less appealing to me.

Technologies significantly locked under the control of private corporations are rarely ones with which I can fully identify. Yes, I might still use those tools of necessity, or after weighing technical trade-offs, but I will never feel membership in their community in the same way.

Sadly also, many projects in the world of open source are infected with a Silicon Valley tech-bro libertarian ideology. While the open source aspect is good, and while communities are something more than their founding individuals, the toxic beliefs of core contributors can often leak into a much less inclusive attitude in the ensuing community.

I hope that the world outside of my US lens is much less the victim of a worship of laissez-faire and belief in a a fantasized meritocracy. I'm here in a place I have not previously been, and perhaps I can learn from those here what values, experiences, hopes, and social structures my colleagues here experience, and what you hope for.

Python does not suffer these pitfalls. Or at least it usually does not. Or at least in its best aspects it does not.



Python as a Social Fact

Institutionalizing Inclusivity

PyLadies is an international mentorship group for marginalized genders, such as but not limited to non-binary people, trans people, and women in tech, with a focus on helping our community become active participants and leaders in the Python open-source community.

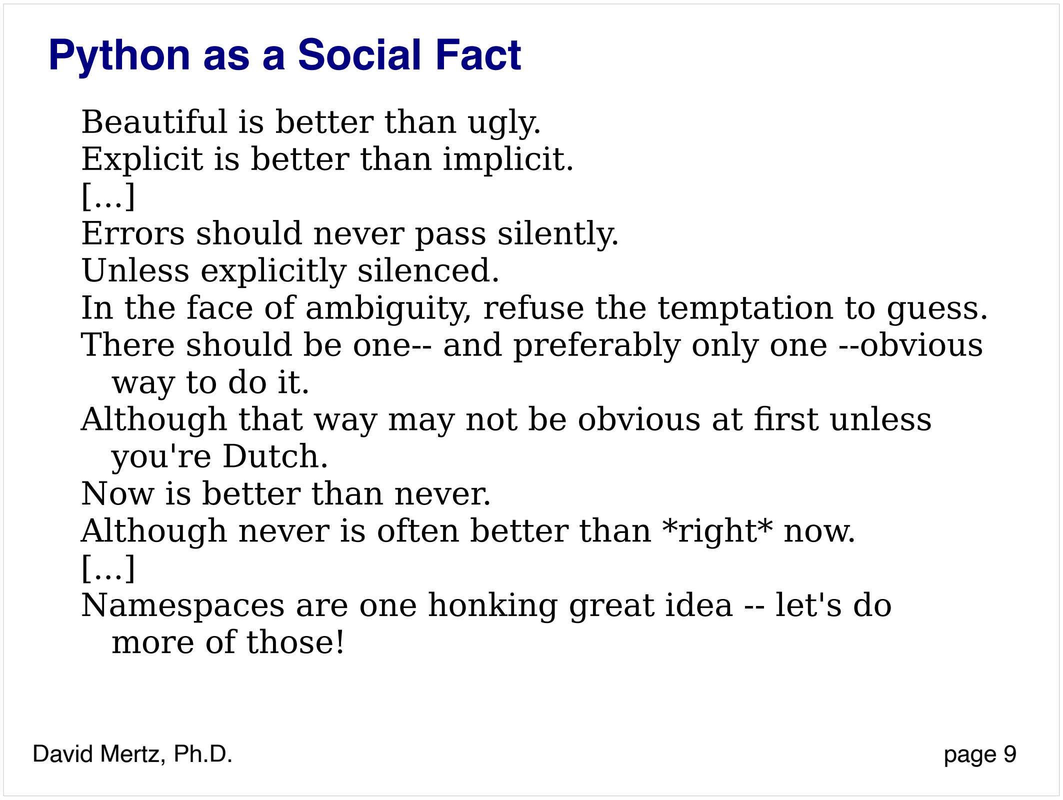
I was present for several of the “doctrine shaping” changes that have led to our current Python communities. I did not realize at the time how important these apparently small actions actually were, but I am delighted by their outcomes.

The next few things I’ll say will skew towards the role of the Python Software Foundation as the nominal “official administrative arm” of the Python community. I recognize that this is limited; one particular US nonprofit is not the same thing as diverse and flourishing communities in many nations, regions, cities, technical areas, and social backgrounds.

By chance I knew the initial founders of PyLadies, in 2008, who were in Los Angeles where I lived at the time. I helped smooth over some initial personality conflicts in that first chapter. A bit later I helped bring the brand and the organization under PSF fiscal sponsorship.

PyLadies has gone on to have hundreds of chapters and events, and so many wonderful members of our, and its, community have promoted equality and opportunity. PyLadies has become an institutional part of what it means to use and belong to Python, whether or not you as an individual attend or organize its events.

The presence of that *doctrinal trend* is part of our identities in belonging to the Python community.



Python as a Social Fact

Beautiful is better than ugly.
Explicit is better than implicit.
[…]
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
[…]
Namespaces are one honking great idea -- let's do more of those!

David Mertz, Ph.D. page 9

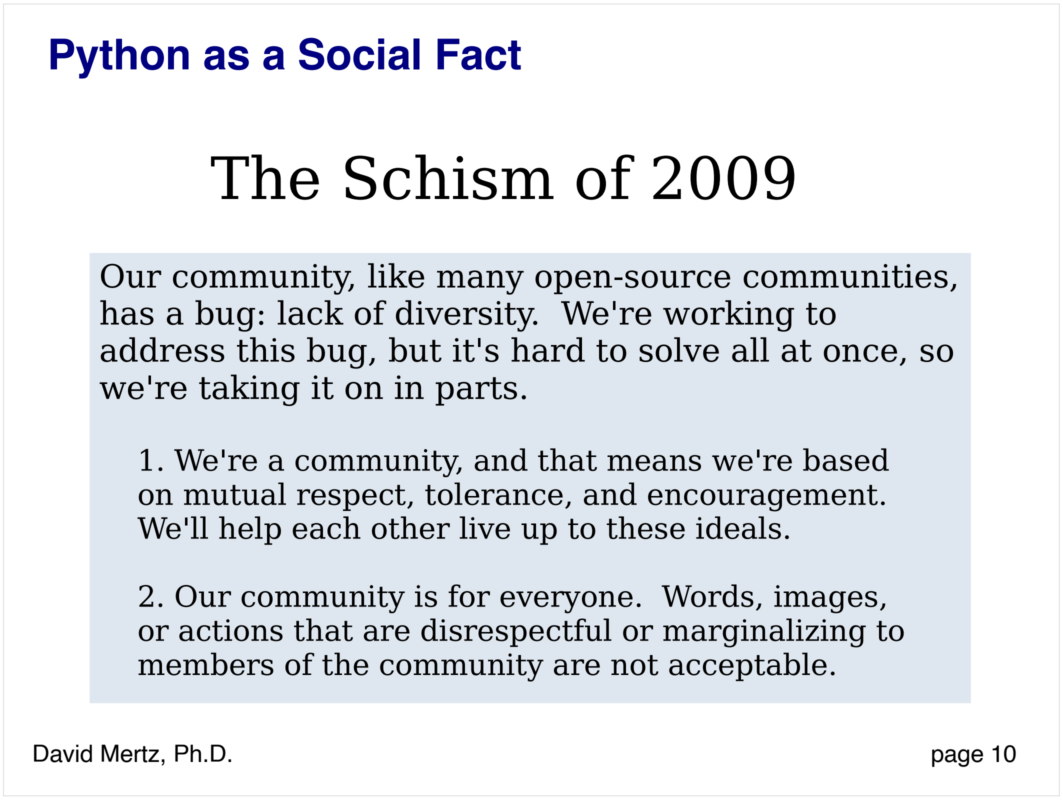
Our dogma is infused with humor. Tim Peters' *Zen of Python* is rich with ideas about what makes our language what it is, but it also "breaks the fourth wall" and has a literary richness. We can hear the winks when reading it.

The naming of projects, companies, libraries, and the like, often carry this joking quality, and is much of what many of us find joy in, within these communities of Python.

Only during writing this talk did I find a comment on a bug tracker, by Tim, from that same year of 2008 that I am treating as a transition. A contributor suggested that the inconsistent use of dashes in the Zen was a mistake. Tim replied:

> I'm afraid you missed the joke ;-) While you believe spaces are required on both sides of an em dash, there is no consensus on this point. For example, most (but not all) American authorities say /no/ spaces should be used. That's the joke. In writing a line about "only one way to do it", I used a device (em dash) for which at least two ways to do it (with spaces, without spaces) are commonly used, neither of which is obvious -- and deliberately picked a third way just to rub it in.

Tim is only partially right, for what it's worth. While indeed em dashes should have no space around them, en dashes always should; and neither should be confused with a hyphen.



Python as a Social Fact

The Schism of 2009

Our community, like many open-source communities, has a bug: lack of diversity. We're working to address this bug, but it's hard to solve all at once, so we're taking it on in parts.

1. We're a community, and that means we're based on mutual respect, tolerance, and encouragement. We'll help each other live up to these ideals.

2. Our community is for everyone. Words, images, or actions that are disrespectful or marginalizing to members of the community are not acceptable.

Python's communities have been self-conscious of what it means to be a community. But we have not been univocal.

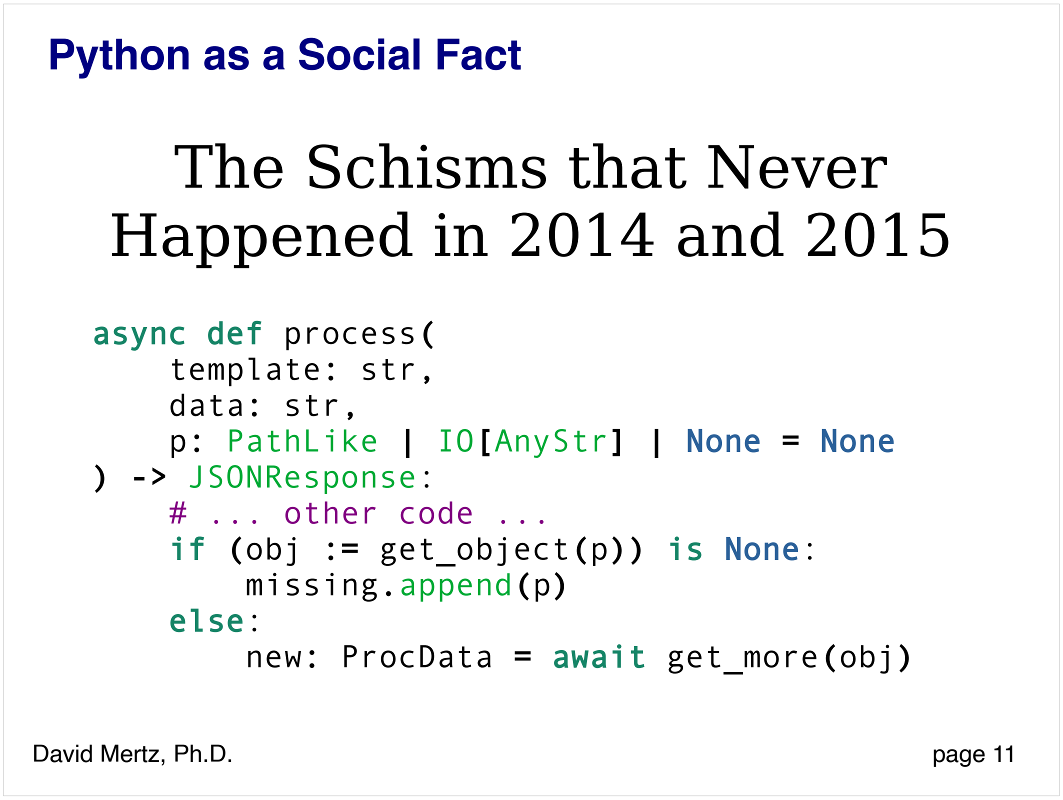
In 2009, the PSF's Diversity Statement—followed by our Code-of-Conduct—were quite divisive. Many Python users and developers angrily resisted an enshrinement of diversity and inclusion. Some left altogether.

In their minds—and I think I can reasonably characterize many of those prominent voices since many have been my close friends—it was wrong to make Python about social values, and only right to concern ourselves with technical merits and language or library design choices.

I supported and helped author the Diversity Statement and CoC that is now used in all Python-branded conferences. While it is fortunately rare that the CoC needs to be used, I believe its presence has contributed to inclusivity.

I believe that some of my friends were wrong ideologically and politically, and I'm proud to have voted as a PSF Director to enact these foundational *social documents*.

Many other technical and programming language communities have followed similar principles to those that have created the Python community; but Python was one of the first to get there.



Python as a Social Fact

The Schisms that Never Happened in 2014 and 2015

```python
async def process(
    template: str,
    data: str,
    p: PathLike | IO[AnyStr] | None = None
) -> JSONResponse:
    # ... other code ...
    if (obj := get_object(p)) is None:
        missing.append(p)
    else:
        new: ProcData = await get_more(obj)
```

David Mertz, Ph.D.

page 11

Social structures are not fixed over time, but to some degree Python's ethical obligations to strive for inclusivity was decided in 2008 and 2009, and this has become a persistent aspiration of our communities.

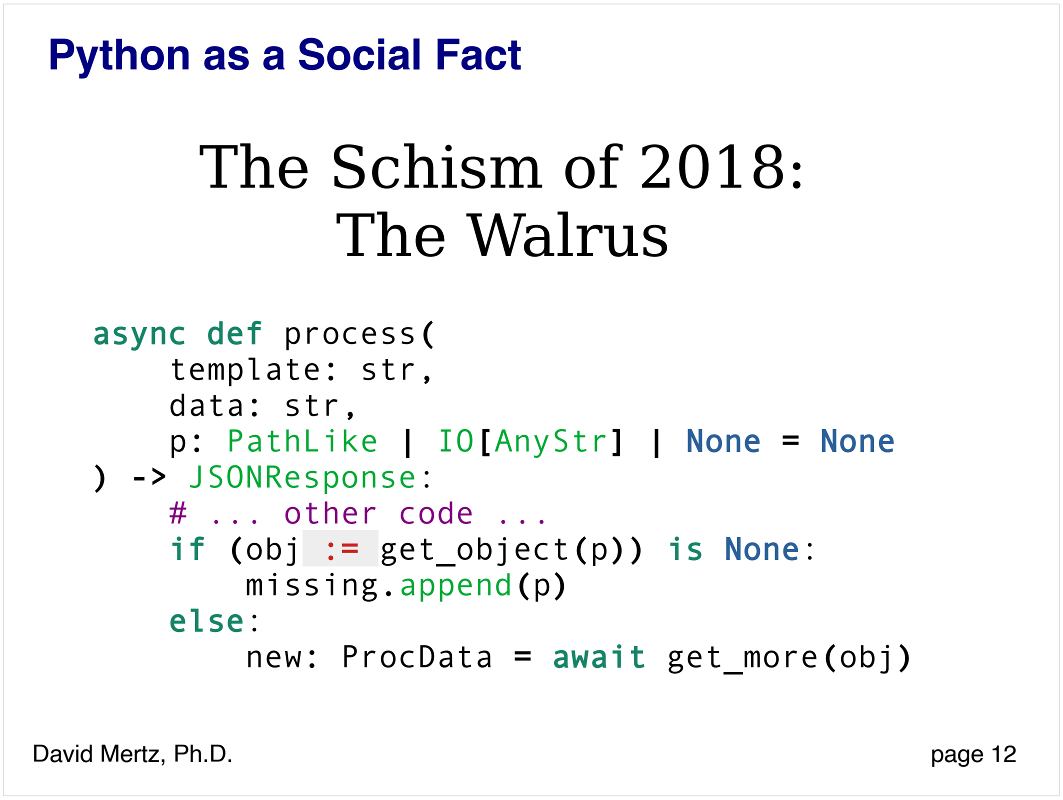
This attitude, albeit imperfectly enacted and sometimes blind to wider global realities outside a privileged technocratic elite, has endured.

But Python *is*, after all, a *programming language* too. Sometimes technical issues divide people as well. The code I show on this slide looks a great deal different than anything I could have written in 2003 or 2008. It's more-or-less a snippet of code I genuinely wrote in the last few months.

In 2014, type annotations were added by PEP 484. In 2015, coroutines spelled with *await* and *async* were introduced. Both of these are large changes, and both occupy a large part of the "code surface" in the snippet you see.

Many Python developers—to some extent including me—have consciously steered around using type annotations and/or asynchronous programming. In some ways, the Python community has become divided along these technical lines, creating sub-communities for different programming styles and tasks.

Such fracturing happens as well around, for example, Django, which is almost a different language embedded in Python, or equally for NumPy, and for Pandas or Polars, each of which fulfill different needs and require different ways of thinking.



Python as a Social Fact

The Schism of 2018: The Walrus

```
async def process(
    template: str,
    data: str,
    p: PathLike | IO[AnyStr] | None = None
) -> JSONResponse:
    # ... other code ...
    if (obj := get_object(p)) is None:
        missing.append(p)
    else:
        new: ProcData = await get_more(obj)
```

This ordinary fragment of code remains in this slide. It contains the "walrus operator," that little colon-equal to allow for assignments inside larger statements.

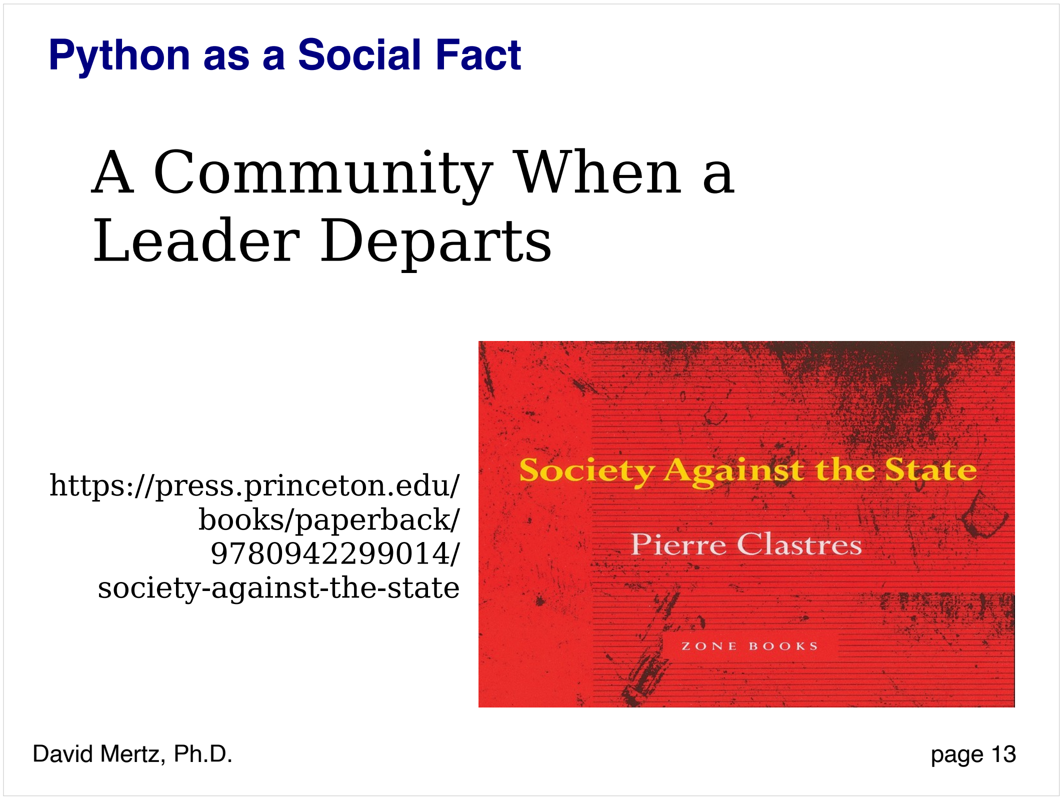
Most of the Python community has given little thought to these two punctuation symbols. Perhaps you use them, perhaps you do not. They probably feel like an incidental matter of syntax, about which your feelings are not overwhelming.

The discussion over the addition of this syntax was as heated, and generated nearly as deep feelings, as the schism of 2008.

While the negative tone of much of this discussion reached relatively few people, it ultimately proved a tipping point for Guido van Rossum, creator of Python and the person long given the joking sobriquet "Benevolent Dictator for Life."

After the discussion of the walrus operator—PEP 572—Guido chose, for his own mental well being, to step down from his privileged leadership role in the Python language itself. He remains part of our community, and remains active in development; but governance of the language has become more strictly procedural.

What did this change in formal structure mean for the global *community* that is only loosely connected to the core contributors to the language?



Python as a Social Fact

A Community When a Leader Departs

https://press.princeton.edu/books/paperback/9780942299014/society-against-the-state

Society Against the State
Pierre Clastres
ZONE BOOKS

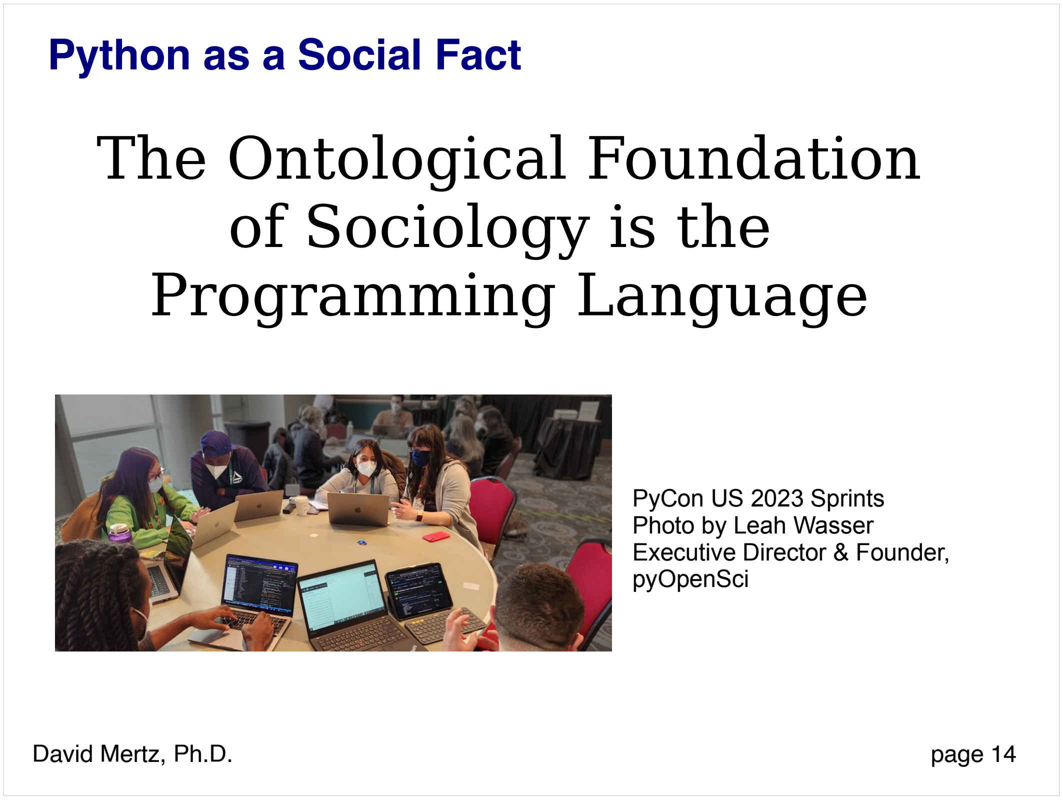
In a Pythonic spirit, my slides are somewhat tongue-in-cheek. Pierre Clastres' 1974 book, *Society Against the State*, is a brilliant critique of conventional Western anthropology which found it acceptable to talk about "primitives" and of linear progression towards more hierarchical societies.

For Clastres, non-hierarchical societies have always been *aware* of hierarchy, and always specifically resisted it rather than foreshadowed it. I earnestly believe that one of the best things each of us can do to create a better Python community—and indeed a better world—is to work wherever we can to resist and undermine hierarchies.

The change in Python's development and governance was not, in reality, an overthrow or rejection of the BDFL. But it was a change in doctrine nonetheless. Whereas we previously found security in the continuing good judgment of a single well-meaning individual, with excellent intuitions about programming language design, something else became necessary.

That something was not particularly non-hierarchical, but maybe just slightly less so than previously. A group called the Python Steering Council was created, and elected yearly by a limited group of those people who are core contributors to CPython. This rotating group of 5 people must, between them, find the same wisdom in their decisions as did the BDFL for Python's first 27 years.

Many of the people in this audience do not know the brief history I've described over the last few minutes. The fact you can all be Pythonistas and have no need to know this actually says something about the Python community—or, better, the many Python communities across the world.



The Ontological Foundation of Sociology is the Programming Language

PyCon US 2023 Sprints
Photo by Leah Wasser
Executive Director & Founder,
pyOpenSci

It is not uncommon for technical communities to dissolve when their founding figures no longer lead. Python is far from alone in staying consistent through such changes, but it's not a simple given.

In authentically open source projects, sometimes this is reflected in a fork of a project, and sometimes by simple abandonment. In corporate controlled software, even where it has an open source license, communities often die at the whims of product managers or VC capital.

What led me to this talk was thinking about Émile Durkheim and later sociologists who were inspired by him. Arguably the central concept in Durkheim's work was the idea that communities are a proper object of study in themselves. Things are true and false of communities as such, and communities have histories and values that are not reducible to the beliefs or psychologies of their members.

The title on this slide is another silly joke, of course. Neither sociology nor programming languages are so reducible nor so close to each other as my provocative lie.

However, I genuinely to think that a programming language is every bit as much a relevant focus for sociological analysis as is any other collection of people, embedded and enshrined in specific structures of governance, belief, and doctrine.

Python as a Social Fact

A Darker Side of Social Cohesion

As much as I truly admire the good in our communities, the very things that make them best are prone to pitfalls of group cohesion. As with other social groups, demands for inclusion can sometimes replace decency and respect for others rather than enhance it.

Codes of conduct are meant to encourage cooperation and deference, but they can sometimes simultaneously attract zealots devoted to their draconian and capricious enforcement. I've cynically characterized these members of the Python community as *Torquemada fanbois*, after the gruesome Grand Inquisitor of the Spanish Inquisition. While Monty Python jokes are a staple of the Python community, this dig comes with discomfort rather than easy humor.

The reasons obviously involve specific individuals, but they can genuinely not be reduced to the motivations and actions of those people. A descent into a miniature authoritarianism can emerge from a collective feeling of what a community means. Those who carry out "purification" generally believe they are acting from consensus, and even from demands of the larger groups.

This specific negative social trend has largely destroyed the Open Source community around the NixOS Linux distribution, and has been endemic within the Scala community, it is sadly prevalent around the Django Software Foundation, and exists in other Free Software communities as well.

Regrettably, some specific discussions and conflicts within the PSF sphere, ongoing over the last few weeks, about removing "undesirable elected Fellows" has followed this course. As I speak, I do not know what the outcome of this schism will be.

Python as a Social Fact

An Anecdote of Foolishness

pyfound.blogspot.com/2015/04/pycon-2018-2019-in-havana-cuba.html

David Mertz, Ph.D.

page 16

Let me tell you a story that is both personal and about power dynamics. In 2015, my life partner was a blogger for the PSF. For April Fool's day of that year—a day where spoofs and gags are ritually prevalent in media and personal interactions—she and I made such a joke for the Python blog. Another example to note, in the Python world, is PEP 401, named after the date of April 1.

In this brief story, we announced that the North American PyCon would next be scheduled in Havana Cuba. The bite of this joke requires an understanding of the enormous hostility US governments have maintained toward Cuba since its revolution, and the details of that are a very different long talk. However, the deadpan presentation was largely about what a good thing it *would be* if that actually happened.

This brief post prompted quite a bit of indignant outrage and hostility towards me. Notably this outrage—which was very much in the spirit of misguided CoC complaints—came entirely from privileged white men in the United States. Mind you, I am a great deal like them demographically, and I advise everyone never to trust people who look or sound like me. But equally notably, my actual Cuban friends and acquaintances *did not* object to the joke, and felt it a positive ironic gesture.

One result of this incidental post was encouraging actual Cuban Pythonistas to organize a conference that was planned to be named PyCon Cuba, although they arrived at a different name in the end. I was absolutely delighted to be one of the speakers at the conference that resulted, and I helped in a small way in growing that community. I most certainly made and retain friendships from that wonderful visit.

I also retain enemies from the post itself. Many are the same people whom I criticized in the last slide.

Python as a Social Fact

An Archipelago of Communities

David Mertz, Ph.D.

page 17

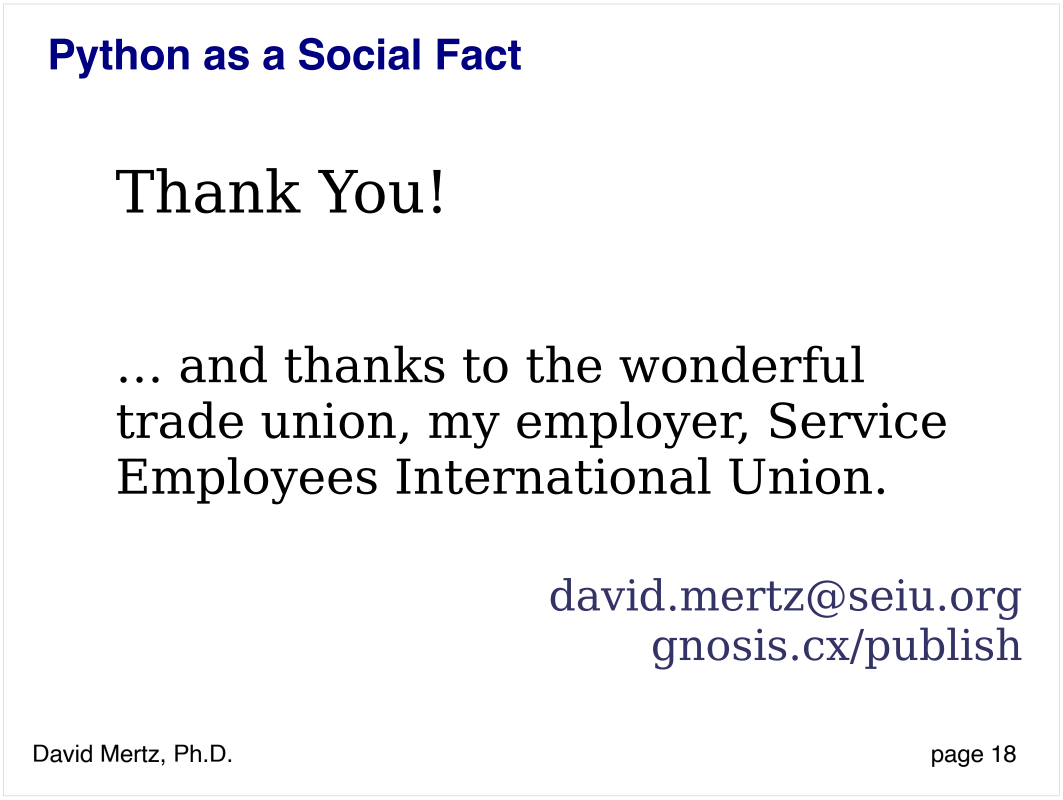
I know in this talk that I have focused too much on the goings-on of the Python Software Foundation, a community of volunteer bureaucrats, of which I am one. For some reasons of modest budgets and grants, and because of some limited intellectual property control, the PSF is important.

However, the PSF is not **that** important. This community of PyCon Nigeria, and those of many other national and regional conferences, user groups, project collectives, and other associations of developers who love this language, are every bit as important to the future of Python as are a handful of paid staff in the United States, or even than the slightly larger collection of Directors who—happily—now mostly come from the Global South.

There are many continuities among our many communities, but there is also a distinctness in what each one brings. We share values around the world, but we also share differences, and much of our theology is to recognize and acknowledge both our commonalities and our variability.

I painted a possible dark portrait a couple slides ago. Indeed it happens, many times in history, that a group of ill-meaning—or still worse, a group of too vehemently well-meaning—people come to control the hierarchy of broad communities. That is unfortunate when it happens.

Even if the worst outcomes were to occur at that far-away originary core, it would only somewhat affect the flourishing peripheries of the Python world. People in this very room could fork the Free Software language, and people throughout the world would continue to feel joy in this shared avocation, profession, devotion, and capacity that Python—and we people who care for each other—bring to our communities.



Python as a Social Fact

Thank You!

… and thanks to the wonderful trade union, my employer, Service Employees International Union.

david.mertz@seiu.org
gnosis.cx/publish

Open source communities—and Python especially—reveal a liberatory and Utopian possibility for belonging. Of course people genuinely do like the syntax, semantics, approach to concurrency, or to mutability, and so on, of a language. Theology doesn’t vanish, but practice is tantamount.

"Social facts" are always also cogent, and identities as, e.g. a "Pythonista," are often deliberately chosen identities. The values embodied in a community both attract people to that community and shape those who belong to it.