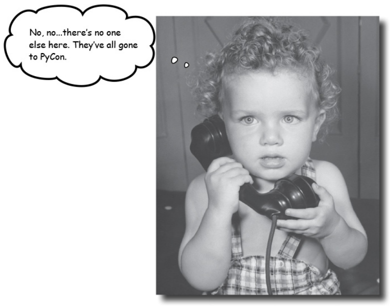
Head First Python, 2nd Edition

PREV
D. Top Ten Projects not Covered: Even More Tools, Lil

NEXT
Index

Appendix E. Getting Involved: The Python Community


No, no...there's no one else here. They've all gone to PyCon.

Python is much more than a great programming language.

It's a great community, too. The Python Community is welcoming, diverse, open, friendly, sharing, and giving. We're just amazed that no one, to date, has thought to put that on a greeting card! Seriously, though, there's more to programming in Python than the language. An entire ecosystem has grown up around Python, in the form of excellent books, blogs, websites, conferences, meetups, user groups, and personalities. In this appendix, we take a survey of the Python community and see what it has to offer. Don't just sit around programming on your own: **get involved**!

BDFL: Benevolent Dictator for Life

Guido van Rossum is a Dutch programmer whose gift to the world is the Python programming language (which he started as a "hobby" in the last 1980s). The ongoing development and direction of the language is set by the *Python core developers*, of which Guido is but one (albeit a very important one). Guido's title of *Benevolent Dictator for Life* is in recognition of the central role he continues to play in the day-to-day life of Python. If you see the letters BDFL in relation to Python, that's a reference to Guido.

Guido is on the record as stating that the name "Python" is a nod (and a wink) toward the British television comedy troupe *Monty Python's Flying Circus*, which helps explain the use of the name `spam` for many of the variables referred to in the Python docs.

Despite Guido's leading role, he does **not** own Python: nobody does. However, the interests of the language are protected by the PSF.

PSF: THE PYTHON SOFTWARE FOUNDATION

The PSF is a nonprofit organization that looks after the interests of Python, and is run by a nominated/elected board of directors. The PSF promotes and sponsors the continued development of the language. This is from the PSF's mission statement:

> *the growth of a diverse and international community of Python programmers.*

Anyone can join the PSF and get involved. See the PSF website for details:

> ***https://www.python.org/psf/***

> **Have your say: join the PSF.**

One of the PSF's major activities is involvement in (and the underwriting of) the annual Python conference: *PyCon*.

PYCON: THE PYTHON CONFERENCE

Anyone can attend (and speak at) PyCon. In 2016, Portland, Oregon, hosted the conference, with thousands of Python developers in attendance (the previous two PyCons were held in Montreal, Canada). PyCon is the largest Python conference, but not the only one. You'll find Python conferences across the globe, ranging in size from small, regional conferences (tens of attendees), through national conferences (hundreds of attendees), up to the likes of *EuroPython* (thousands of attendees).

> **Get involved: attend PyCon.**

To see if there's a PyCon near you, search for the word "PyCon" together with the name of your nearest city (or the country you live in). Chances are, you'll be pleasantly surprised by what you find. Attending a local PyCon is a great way to meet and interact with like- minded developers. Many of the talks and sessions at the various PyCons are recorded: pop over to *YouTube* and type "PyCon" for an idea of what's available to view.

A Tolerant Community: Respect for Diversit

Of all the programming conferences that exist today, PyCon was one of the first to introduce and insist on a *Code of Conduct*. You can read the 2016 Code of Conduct here:

> ***https://us.pycon.org/2016/about/code-of-conduct/***

Such a development is a *very good thing*. More and more, the smaller regional PyCons are adopting the Code of Conduct, too, which is also very welcome. A community grows to be strong and inclusive when there are clear guidelines about what's acceptable and what isn't, and the Code of Conduct helps to make sure all the world's PyCons are as welcoming as they can be.

In addition to striving to ensure everyone is welcome, a number of initiatives attempt to increase the representation of specific groups within the Python community, especially where—traditionally—such groups have been underrepresented. The best-known of these is *PyLadies*, which was established per their mission to help "more women become active participants and leaders in the Python open source community." If you're lucky, there's a *PyLadies* "chapter" near you: find out by starting your search from the *PyLadies* website:

> ***http://www.pyladies.com (http://www.pyladies.com)***

Just like the Python community, *PyLadies* started out small, but has very quickly grown to have global reach (which is truly inspirational).

> **Encourage and support diversity within the Python community.**

COME FOR THE LANGUAGE, STAY FOR THE COMMUNITY

Many programmers new to Python comment on how inclusive the Python community is. A lot of this attitude stems from Guido's guiding hand and example: firm, yet benevolent. There are other leading lights, too, and plenty of inspirational stories.

It doesn't get much more inspirational than *Naomi Ceder's* talk at *EuroPython* (which was repeated at other regional conferences, including *PyCon Ireland*). Here's a link to Naomi's talk, which we encourage you to watch:

> ***https://www.youtube.com/watch?v=cCCiA-IlVco***

Naomi's talk surveys a life in Python, and discusses how the community supports diversity, and how there's always more work for everyone to do.

One way to learn more about a community is to listen to some of the podcasts generated by its participants. We discuss two Python podcasts next.

Python Podcasts

There are podcasts on *everything* these days. Within the Python community, there are two we feel are well worth subscribing and listening to. Whether it's something to listen to while driving, cycling, running, or chilling out, these podcasts are both deserving of your attention:

- *Talk Python to Me*: ***https://talkpython.fm***
- *Podcast.__init__*: ***http://pythonpodcast.com (http://pythonpodcast.com)***

There's nothing quite like working out to the Python-related podcasts.

Follow both of these podcasts on *Twitter*, tell your friends about them, and give the producers of these podcasts your full support. Both *Talk Python To Me* and *Podcast.__init__* are produced by regular members of the Python community for the benefit of all of us (and *not* for profit).

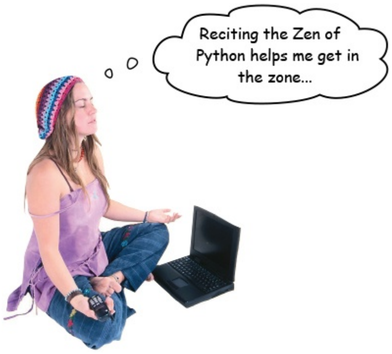
PYTHON NEWSLETTERS

If podcasts aren't your thing, but you still want to keep up with what's happening in the Python world, there are three weekly newsletters that can help:

- Pycoder's Weekly: ***http://pycoders.com*** *(http://pycoders.com)*
- Python Weekly: ***http://www.pythonweekly.com*** *(http://www.pythonweekly.com)*
- Import Python: ***http://importpython.com/newsletter*** *(http://importpython.com/newsletter)*

These curated newsletters provide links to all types of material: blogs, vlogs, articles, books, videos, talks, new modules, and projects. And their weekly announcements arrive right to your email inbox. So, go ahead and sign up.

As well as a foundation, multiple conferences, subgroups like *PyLadies*, codes of conduct, recognition of diversity, podcasts, and newsletters, Python also has its very own notion of *Zen*.



The Zen of Python

Many moons ago, Tim Peters (one of Python's early leading lights) sat down and wondered: *what is it that makes Python Python?*

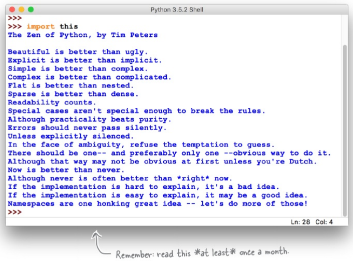
The answer came to Tim as *The Zen of Python*, which you can read by starting any version of the interpreter and typing the following incantation into the `>>>` prompt:

```
import this
```

We've done this for you, and shown the output of the above line of code in the screenshot at the bottom of this page. Be sure to read *The Zen of Python* at least once a month.

Many have tried to compress *The Zen of Python* into something a little easier to digest. None other than `xkcd` has given it a go. If you're connected to the Internet, type this line of code into your `>>>` prompt to see (quite literally) how `xkcd` got on:

```
import antigravity
```



```
>>>
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
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.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

Remember: read this *at least* once a month.

Which Book Should I Read Next?

Is that it? You can't finish off without giving me some recommendation on what I should read next.

Our Favorite Python Books

As Python has grown in popularity, the number of books devoted to the language has blossomed. Of all the books out there, there are two we regard as indispensable.

Recipes for Mastering Python 3
3rd Edition
Python Cookbook
REILLY
David Beazley & Brian K. Jones
O'REILLY
Fluent Python
CLEAR, CONCISE, AND EFFECTIVE PROGRAMMING
Luciano Ramalho

We mentioned David Beazley's work in an earlier appendix. In this book, David teams up with Brian K. Jones to document a wonderful collection of Python coding recipes. If you find yourself wondering how you do something in Python, wonder no more: look up the answer in Python Cookbook.

If deep-dives are more your thing, read this excellent book. There's a lot in here, but it's all good (and you'll be a better Python programmer for the experience).

PREV
D. Top Ten Projects not Covered: Even More Tools, Libraries, an…

NEXT
Index