The Zen of Python in Practice
Balancing Idealism With Production Concerns
The Beautiful Dream
import this
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!
Readability Counts

PEP8

ALL THE THINGS!!!
(but so does the blame line)
There should be one--and preferably only one--obvious way to do it.
Inter-team dependencies and crazy edge cases
What now?
Zen and the Art of Legacy Code Maintenance
You can't clean code you don't understand
If it works, don't poke it.
Ask questions, but don't insult your predecessors
Code maintenance is kindergarten soccer--a team sport and we don't keep score
Public park rules – leave things nicer than when you arrived
Cuss up a storm in your comments
Any guesses as to what this is?
Fun Fact

Even the Zen of Python source code violates the Zen of Python

You cannot tell me that this is not needlessly complicated 😞
Thank you!

Slides available at tiny.cc/zopip