

What's New in Python 3.14

PyCon Portugal



@anze3db, July 24, 2025

Slides and examples

```
> wc -w Doc/whatsnew/3.1*  
11325 Doc/whatsnew/3.10.rst  
12198 Doc/whatsnew/3.11.rst  
11241 Doc/whatsnew/3.12.rst  
12991 Doc/whatsnew/3.13.rst  
14449 Doc/whatsnew/3.14.rst  
2399 Doc/whatsnew/3.15.rst
```

14449 words = ~2h @ 130wpm

PEP 779: Free-threaded Python

PEP 734: Multiple interpreters

PEP 750: Template strings

(Contributed by Jim Baker, Guido van Rossum, Paul Everitt, Koudai Aono, Lysandros Nikolaou, Dave Peck, Adam Turner, Jelle Zijlstra, Bénédict Tran, and Pablo Galindo Salgado in [gh-132661](#).)

PEP 768: Safe external debugger

(Contributed by Pablo Galindo Salgado, Matt Wozniski, and Ivona Stojanovic in [gh-131591](#).)

Asyncio introspection

(Contributed by Pablo Galindo, Łukasz Langa, Yury Selivanov, and Marta Gomez Macias in [gh-91048](#).)

PEP 784: Zstandard in stdlib

(Contributed by Emma Harper Smith, Adam Turner, Gregory P. Smith, Tomas Roun, Victor Stinner, and Rogdham in [gh-132983](#).)

PEP 758: Allow `except` and `except*` expressions without parentheses

(Contributed by Pablo Galindo and Brett Cannon in [gh-131831](#).)

PEP 649 and 749: Deferred evaluation of annotations

(Contributed by Jelle Zijlstra in [gh-119180](#); [PEP 649](#) was written by Larry Hastings.)

Improved error messages

(Contributed by Pablo Galindo, Tushar Sadhwani, Steele Farnsworth, Sergey Miryanov, Nikita Sobolev, CF Bolz-Tereick and Victor Stinner)

More **colors** everywhere

`python -m unittest`

`python -m json`

`python -m calendar`

PEP 765: Disallow return/break/ continue that exit a finally block

A new type of interpreter

(Contributed by Ken Jin in [gh-128563](#), with ideas on how to implement this in CPython by Mark Shannon, Garrett Gu, Haoran Xu, and Josh Haberman.)

A new type of interpreter

- Tail calls between small C functions for Python opcodes instead of a large C case statement
- geometric mean of ~~9-15%~~ **3-5%** faster
- Clang 19 and newer on x86-64 and AArch64
- opt-in for now

**Binary releases for the
experimental just-in-time compiler**

Binary releases for the experimental just-in-time compiler

- not recommended for production use
- `PYTHON_JIT=1`
- performance impact of enabling it can range from 10% slower to 20% faster

Incremental garbage collection

(Contributed by Mark Shannon in [gh-108362](#).)

Incremental garbage collection

- cycle garbage collector is now incremental
- maximum pause times are reduced by an order of magnitude

Questions?

@anze3db, July 24, 2025



Slides and examples



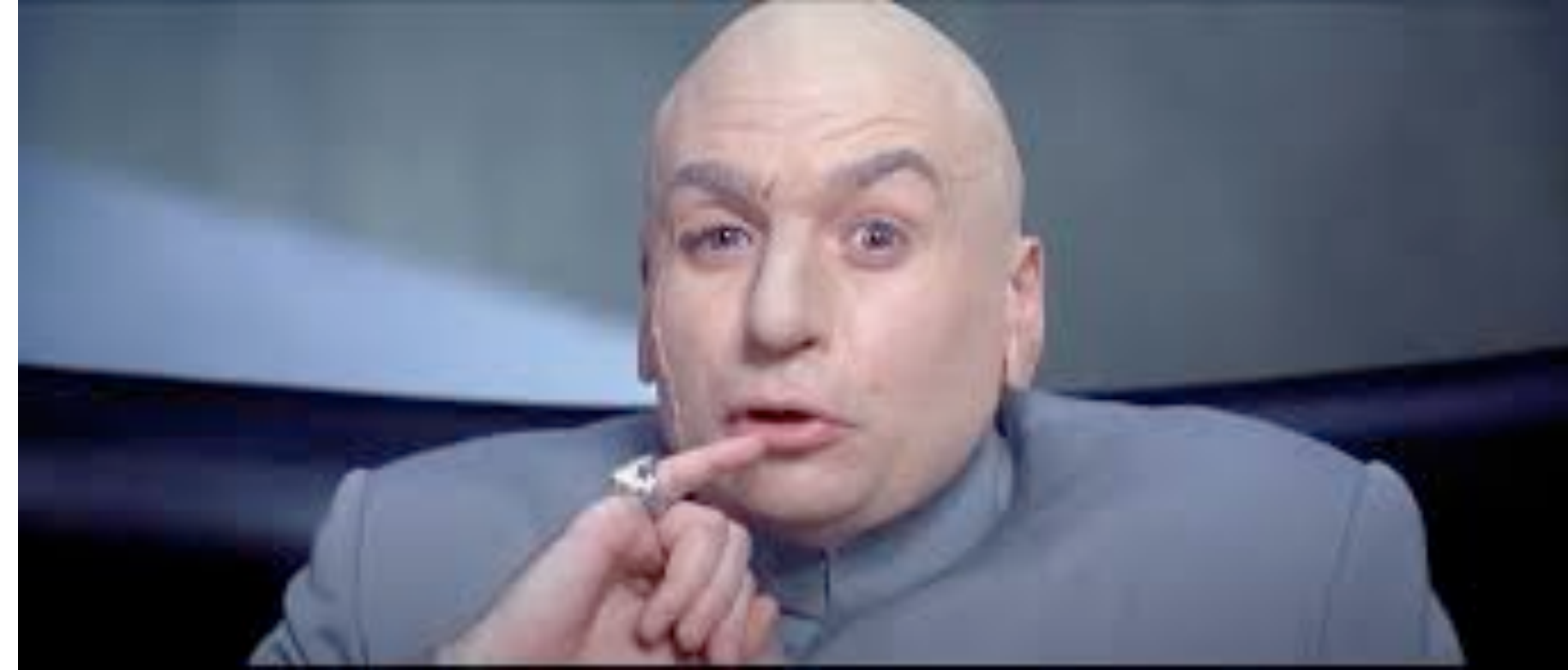
@anze3db

<https://pecar.me>

anze@pecar.me

First big feature for **Python 3.15**

**High frequency statistical
sampling profiler**



Up to 1,000,000 Hz sampling rate

PEP 741: Python configuration C API

(Contributed by Victor Stinner in [gh-107954](#).)

Concurrent safe warnings control

(Contributed by Neil Schemenauer and Kumar Aditya in [gh-130010](#).)