

# Куда двигается python в 2022 году?

Денис Аникин, <https://xfenix.ru>

# **Представляюсь**

**Меня зовут Денис и я...**

— Техлид нескольких команд в Райфе

# **Представляюсь**

**Меня зовут Денис и я...**

- Техлид нескольких команд в Райфе
- Community Lead Python Community тоже в Райфе :)

# Представляюсь

Меня зовут Денис и я...

- Техлид нескольких команд в Райфе
- Community Lead Python Community тоже в Райфе :)
- Fullstack: python + typescript

# Представляюсь

Меня зовут Денис и я...

- Техлид нескольких команд в Райфе
- Community Lead Python Community тоже в Райфе :)
- Fullstack: python + typescript
- Много занимаюсь DevOps

# Представляюсь

Меня зовут Денис и я...

- Техлид нескольких команд в Райфе
- Community Lead Python Community тоже в Райфе :)
- Fullstack: python + typescript
- Много занимаюсь DevOps
- Мой сайт: <https://xfenix.ru>

Перейдём к теме

# **О чем будем говорить?**

**Название у меня опять размытое...**

— Кратко о том, что было интересного на Python Language Summit 2021 и 2022 года



# О чем будем говорить?

Название у меня опять размытое...

- Кратко о том, что было интересного на Python Language Summit 2021 и 2022 года
- О том что из этого уже сделано

# О чем будем говорить?

Название у меня опять размытое...

- Кратко о том, что было интересного на Python Language Summit 2021 и 2022 года
- О том что из этого уже сделано
- И что будет сделано

# О чем будем говорить?

Название у меня опять размытое...

- Кратко о том, что было интересного на Python Language Summit 2021 и 2022 года
- О том что из этого уже сделано
- И что будет сделано
- И, в основном, про наше больное — про скорость!

Slowest things on earth:



# Python Language Summit







# Python Language Summit

— Проводится каждый год где-то в апреле-мае

# Python Language Summit

- Проводится каждый год где-то в апреле-мае
- Мероприятие предназначено для разработчиков cpython, руру, jython и т.д. На нём они делятся проблемами и, возможно, решениями



# Python Language Summit

- Проводится каждый год где-то в апреле-мае
- Мероприятие предназначено для разработчиков cpython, pyru, jython и т.д. На нём они делятся проблемами и, возможно, решениями
- Там собираются все ключевые разработчики и steering council

Cinder

The logo for Cinder, featuring the word "cinder" in a white, lowercase, sans-serif font. A small, stylized orange flame icon is positioned above the letter 'i'. The text and icon are centered within a solid black rectangular background.

cinder

# Cinder

Форк CPython от одной широко известной экстремистской  организации

— Форк python 3.8 известной сети начинающейся на инст,  
заканчивающейся на грам

# Cinder

Форк CPython от одной широко известной экстремистской  организации

- Форк python 3.8 известной сети начинающейся на инст, заканчивающейся на грам
- Ориентирован на производительность

# Cinder

Форк CPython от одной широко известной экстремистской  организации

- Форк python 3.8 известной сети начинающейся на инст, заканчивающейся на грам
- Ориентирован на производительность
- Содержит много интересных доработках

# Cinder

Форк CPython от одной широко известной экстремистской  организации

- Форк python 3.8 известной сети начинающейся на инст, заканчивающейся на грам
- Ориентирован на производительность
- Содержит много интересных доработках
- Доработки апстримятся в основной бранч cpython!

# Интересное в Cinder

— `async def` возвращает результат не через `StopIteration`



# Интересное в Cinder

- `async def` возвращает результат не через `StopIteration`
- eager evaluation

# Интересное в Cinder

- `async def` возвращает результат не через `StopIteration`
- eager evaluation
- inline caching for bytecode

# Интересное в Cinder

- `async def` возвращает результат не через `StopIteration`
- eager evaluation
- inline caching for bytecode
- JIT 🔥 (`asmjit`)

# Интересное в Cinder

- `async def` возвращает результат не через `StopIteration`
- eager evaluation
- inline caching for bytecode
- JIT 🔥 (`asmjit`)
- Strict modules

# Интересное в Cinder

- `async def` возвращает результат не через `StopIteration`
- eager evaluation
- inline caching for bytecode
- JIT 🔥 (`asmjit`)
- Strict modules
- Static python

**NoGIL**

# NoGIL

— Анонсирован в октябре 2021 года

# NoGIL

- Анонсирован в октябре 2021 года
- Основан на python 3.9



# NoGIL

- Анонсирован в октябре 2021 года
- Основан на python 3.9
- Работает!

# NoGIL

- Анонсирован в октябре 2021 года
- Основан на python 3.9
- Работает!
- Предложили закрыть под флаг компиляции

# NoGIL

- Анонсирован в октябре 2021 года
- Основан на python 3.9
- Работает!
- Предложили закрыть под флаг компиляции
- Решения о принятии MR не принято :(



# Множественные интерпретаторы

# Subinterpreters

— Работа ведется уже давно

# Subinterpreters

- Работа ведется уже давно
- Предлагается модуль `interpreters` в PEP 554

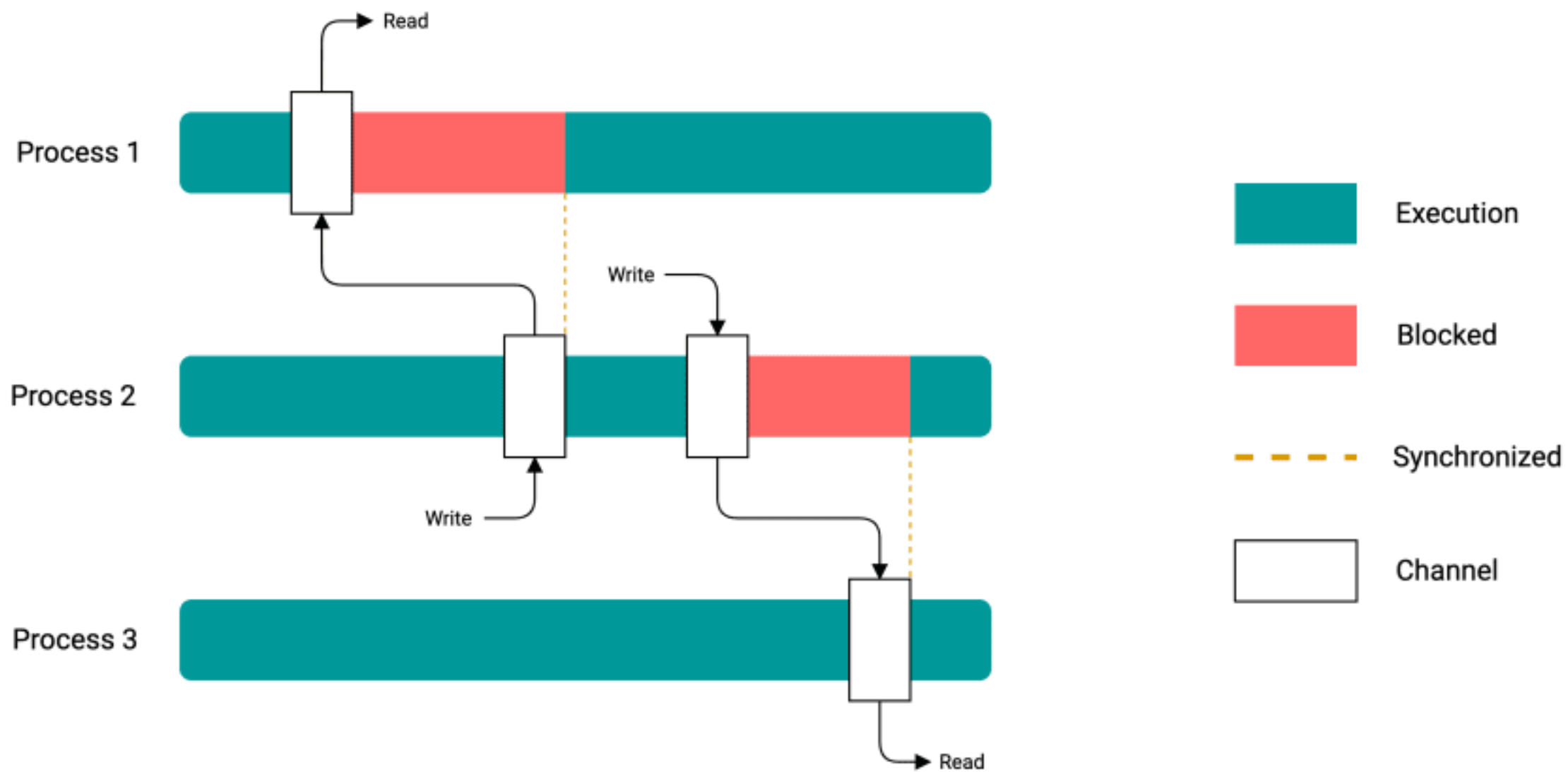
# Subinterpreters

- Работа ведется уже давно
- Предлагается модуль `interpreters` в PEP 554
- По сути дает нам еще одну модель выполнения:  
CSP



# Subinterpreters

- Работа ведется уже давно
- Предлагается модуль `interpreters` в PEP 554
- По сути дает нам еще одну модель выполнения: CSP
- В PEP 684 предлагают каждому такому интерпретатору выделять свой GIL



# Faster CPython

**В начале поговорим чем  
занимался последнее время  
Гвидо**

# Краткая рабочая история

— В октябре 2019 покинул dropbox и ушел «на пенсию» (вспомним о «моржовой битве»)

# Краткая рабочая история

- В октябре 2019 покинул dropbox и ушел «на пенсию» (вспомним о «моржовой битве»)
- В ноябре 2020 года присоединился к Microsoft

# Краткая рабочая история

- В октябре 2019 покинул dropbox и ушел «на пенсию» (вспомним о «моржовой битве»)
- В ноябре 2020 года присоединился к Microsoft
- Получил финансирование от Microsoft и использует его для работы над проектом «faster cpython»

**В 2020 (вероятно) Марк Шеннон,  
Гвидо и Эрик Сноу начали проект  
faster cpython**



# Ключевые факты о «проекте»/команде

- Финансируется Microsoft и Bloomberg

# Ключевые факты о «проекте»/команде

- Финансируется Microsoft и Bloomberg
- Изначально ставил перед собой цель «ускорить питон в 5 раз за 4 года» (к 2025 году)

# Ключевые факты о «проекте»/команде

- Финансируется Microsoft и Bloomberg
- Изначально ставил перед собой цель «ускорить питон в 5 раз за 4 года» (к 2025 году)
- Математика ускорения не отличается хитростью. 4 релиза, каждый из которых ускоряется на 50%:  **$1.5^4 \sim 5$**

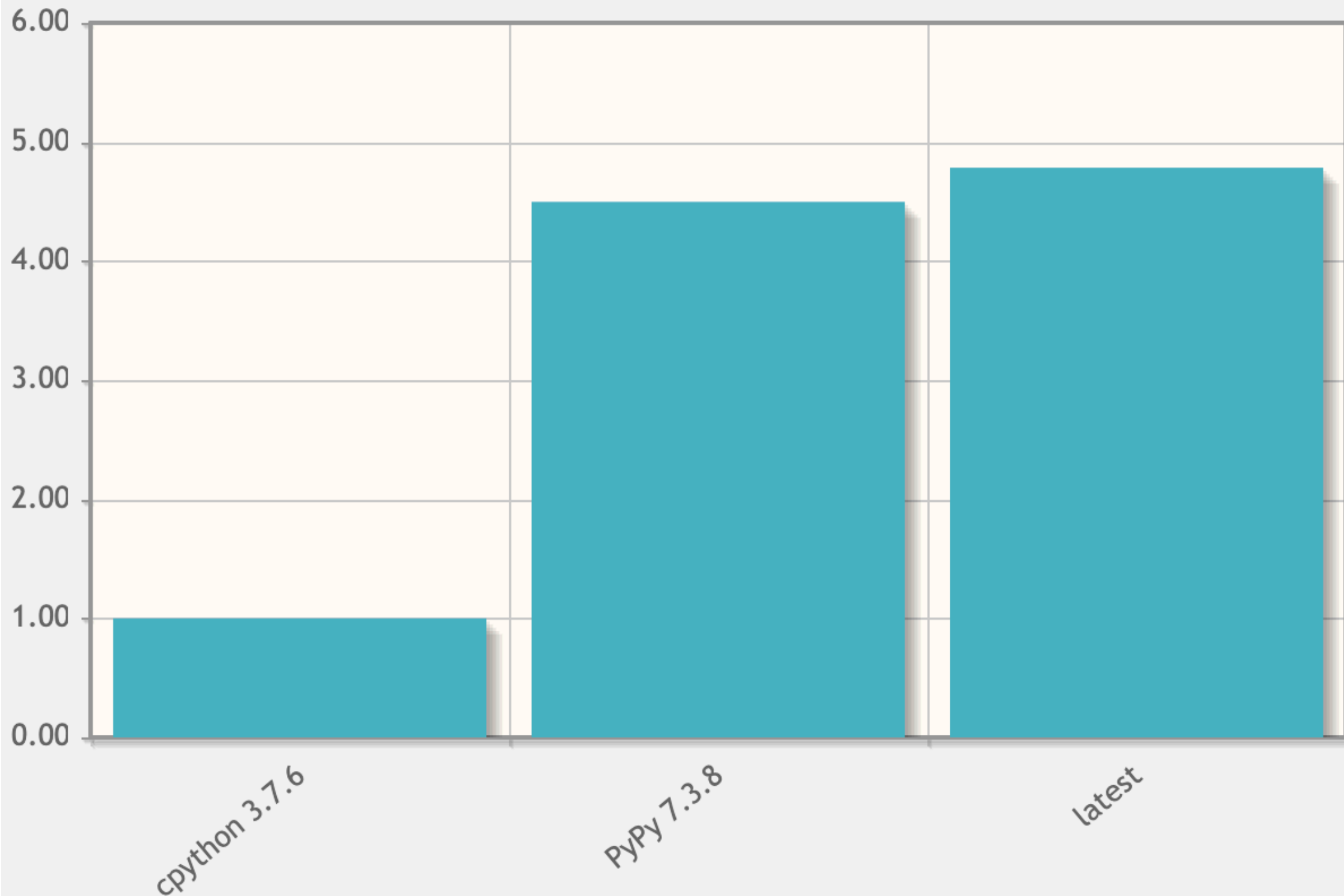
**«Не похоже ли это на  
популизм?»**

**— Спросите вы**

**Да, но в этом ничего  
плохого нет!**

**— Отвечу я**

## How has PyPy3.9 performance evolved over time?



# Как уже ускорился python 3.11?

От **19%** до **25%** в среднем и до 60% в отдельных сценариях относительно python 3.10 (который итак быстрый)

**Как этого достигли?**



# Тут несколько сложно

— Это благодаря «PEP 659 – Specializing Adaptive Interpreter» от Mark Shannon

# Тут несколько сложно

- Это благодаря «PEP 659 – Specializing Adaptive Interpreter» от Mark Shannon
- Если коротко: медленные куски байт-кода заменяются более быстрыми и более специфичными наборами инструкций

**А теперь переходим к «four tiers  
of execution»**

# **Марк Шеннон сформулировал 4 уровня исполнения**

— Tier 0: «прохладный» код, исполняющийся 1 раз, например

# Марк Шеннон сформулировал 4 уровня исполнения

- Tier 0: «прохладный» код, исполняющийся 1 раз, например
- Tier 1: «теплый» код

# **Марк Шеннон сформулировал 4 уровня исполнения**

- Tier 0: «прохладный» код, исполняющийся 1 раз, например
- Tier 1: «теплый» код
- Tier 2: «горячий»

# Марк Шеннон сформулировал 4 уровня исполнения

- Tier 0: «прохладный» код, исполняющийся 1 раз, например
- Tier 1: «теплый» код
- Tier 2: «горячий»
- Tier 3: полыхающий :)

# Марк Шеннон сформулировал 4 уровня исполнения

- Tier 0: «прохладный» код, исполняющийся 1 раз, например
- Tier 1: «теплый» код **← это сделано в 3.11**
- Tier 2: «горячий»
- Tier 3: полыхающий :)



# Марк Шеннон сформулировал 4 уровня исполнения

- Tier 0: «прохладный» код, исполняющийся 1 раз, например
- Tier 1: «теплый» код
- Tier 2: «горячий» ← **здесь полу-jit**
- Tier 3: полыхающий :)

# Марк Шеннон сформулировал 4 уровня исполнения

- Tier 0: «прохладный» код, исполняющийся 1 раз, например
- Tier 1: «теплый» код
- Tier 2: «горячий»
- Tier 3: полыхающий :) **← здесь полный jit**



I GET IT NOW

5x == JIT

**ЛІТ до 3.13  
версії не будет!**

**— По заявлению Марка Шеннона в 2022 году**

**Напоследок немного о  
ближайших планах — python 3.12**

# Планы на 3.12

- Trace optimizer

# Планы на 3.12

- Trace optimizer
- Мультипоточный параллелизм (wow!)



# Планы на 3.12

- Trace optimizer
- Мультипоточный параллелизм (wow!)
- Уменьшение размера некоторых python объектов

# Планы на 3.12

- Trace optimizer
- Мультипоточный параллелизм (wow!)
- Уменьшение размера некоторых python объектов
- Уменьшение оверхеда на управление памятью

# Интересные ссылки

- <https://github.com/python/steering-council>
- [https://pyfound.blogspot.com/2022/05/the-2022-python-language-summit\\_2.html](https://pyfound.blogspot.com/2022/05/the-2022-python-language-summit_2.html)
- <https://lwn.net/Articles/840248/>
- <https://github.com/markshannon/faster-cpython/blob/master/tiers.md>
- <https://github.com/faster-cpython/ideas/wiki/Python-3.12-Goals>
- <https://peps.python.org/pep-0554/>
- <https://peps.python.org/pep-0684/>
- <https://peps.python.org/pep-0659/>

**Спасибо за внимание!**

Денис Аникин, <https://xfenix.ru>