

PYCON  
RUSSIA



ДомКлик

Центр недвижимости  
от Сбербанка

# Cython: программирование на С для людей

Сергей Борисов

## Центр Недвижимости от Сбербанка

- Единый портал недвижимости и ипотечного кредитования Сбербанка
- Быстрая, комфортная и безопасная альтернатива традиционной ипотечной сделке
- Надежные и полезные сервисы
- Дружная и веселая команда
- <http://team.domclick.ru>



## Продукты ЦНС

- Ипотека и неипотечные сделки;
- Витрина объектов;
- Полезные сервисы;
- Телефония с NLP и CV;
- Deep Learning в области обработки изображений



Вместе мы пройдем через...



Let's go. In and out. Twenty minutes adventure.

## Материалы



[https://github.com/juntatalor/pycon\\_2018\\_cython](https://github.com/juntatalor/pycon_2018_cython)

## Нам понадобится

Установленные Cython и Jupyter Notebook

```
pip install Cython jupyter && jupyter notebook
```



Установленный Redis

... или Docker-контейнер с Redis:

```
docker run --rm --name pycon-redis -p 6379:6379 -d  
redis
```



... или сервер на Python с аналогичным интерфейсом

```
python redis_server
```





## Прежде чем начать

*«We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil.*

***Yet we should not pass up our opportunities in that critical 3%.»***

Дональд Эрвин Кнут, сборник «Computing Surveys»  
(Vol. 6, № 4, декабрь 1974, стр. 268).



## Прежде чем начать

- Мониторинг и анализ сетевых запросов
- Уменьшение числа сетевых запросов
- Использование готовых библиотек  
(ujson, uvloop, re2, etc)
- Профилирование кода на Python  
(CProfileV, SnakeViz)





## Подключение внешних библиотек: ctypes & cffi

<https://cffi.readthedocs.io/en/latest/overview.html>

```
import ctypes

from ctypes.util import find_library

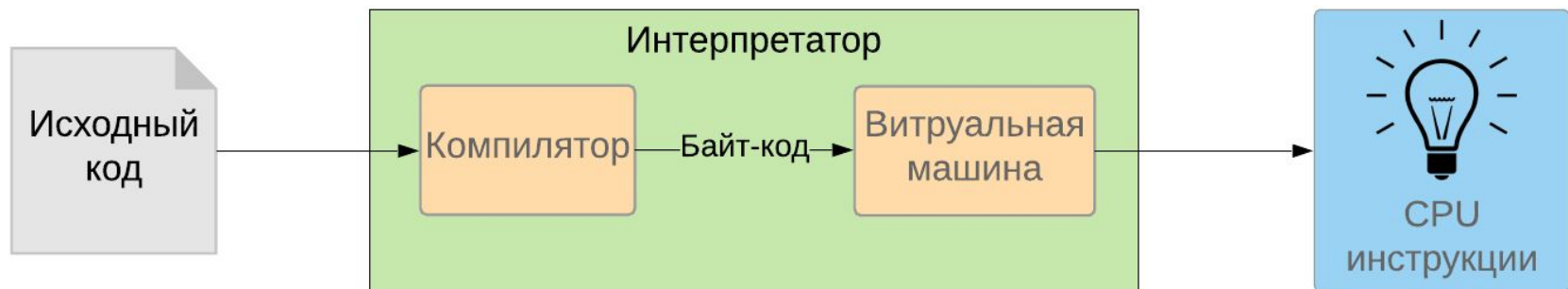
libm = ctypes.cdll.LoadLibrary(find_library("m"))

libm.cos.argtypes = [ctypes.c_double]
libm.cos.restype = ctypes.c_double

def cos_func(arg):
    return libm.cos(arg)
```



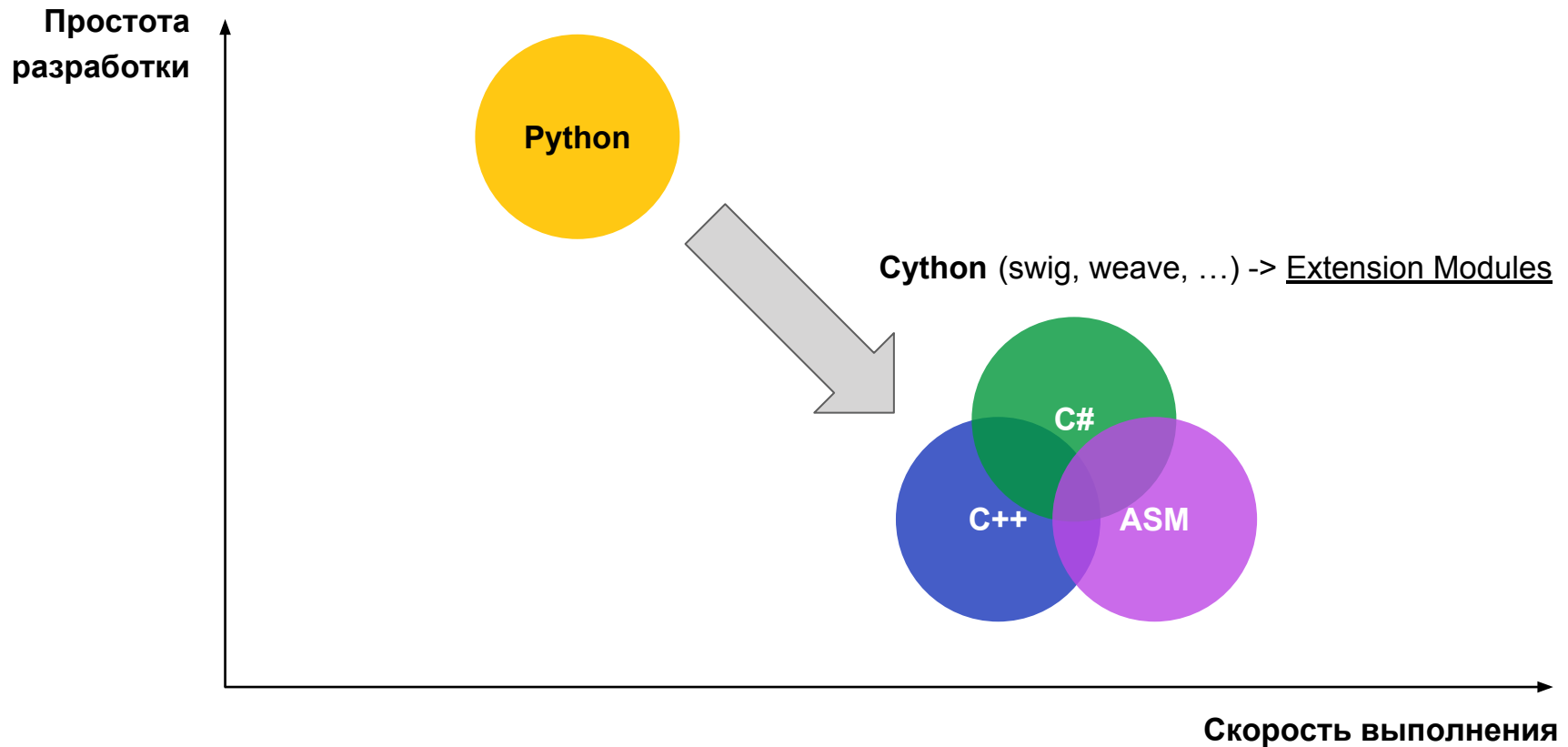
## Почему CPython медленный?





**А можно быстрее?**

## Почему CPython медленный?



## Почему CPython медленный?

```
a, b = 1, 2  
a = a + b
```

- Получение указателей **PyObject\*** на a и b
- Получение функции для сложения нужных типов
- Извлечение примитивов из **PyObject\*** и операция сложения
- Помещение результата в (возможно новый) **PyObject\***

# Python C-API

<https://docs.python.org/3/c-api/intro.html>

```
#include <Python.h>
#include <math.h>

static PyObject* sin_func(PyObject* self,
PyObject* args)
{
    double value;
    double answer;

    if (!PyArg_ParseTuple(args, "d", &value))
        return NULL;

    answer = sin(value);

    return Py_BuildValue("f", answer);
}
```





## Компиляция без Jupyter Notebook

```
import pyximport  
pyximport.install()
```

- Пишем код в файлах с разрешением **\*.pyx**
- Для использования в production используем setup.py



## RESP - это просто

<https://redis.io/topics/protocol>

Типизация по первому байту запроса / ответа:

- Обычные строки без CR (0x0D) LF (0x0A): « + »
- Ошибки: « - »
- Integer: « : »
- Бинарные данные (всегда с указанием размера): « \$ »
- Массивы (всегда с указанием размера): « \* »



## RESP - это просто

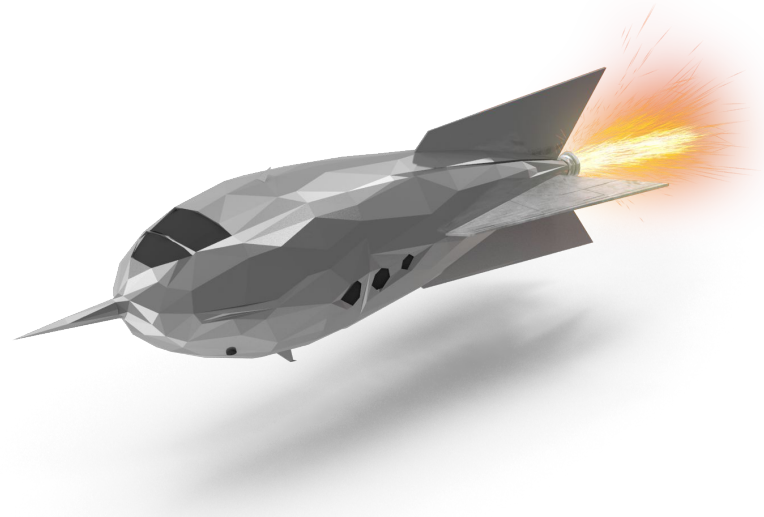
```
* 1 \r\n $ 4 \r\n PING \r\n
```

- Массив
- Из одного элемента
- Элемент - безопасная строка длиной 4 байта
- Команда **PING**

```
+ PONG \r\n
```

- Простая строка
- Ответ PONG

## Ради чего? Benchmark



- Python: **1x**
- Cython без указания типов: **1.3x**
- Cython с указанием типов: **2.7x**



# Спасибо за внимание!



@juntatalor



saborisov@domclick.ru