

Университет ИТМО, кафедра вычислительной техники
Еженедельная отчётная работа по «Информатике»: аннотация к статье

Выполнил(а) Братчиков Иван Станиславович, № группы P3101, оценка
Фамилия И.О. студента не заполнять

| | | |
|--|---|--|
| Название статьи/главы книги/видеолекции Python — это медленно. Почему? | | |
| ФИО автора статьи (или e-mail) @ru_vds | Дата публикации (не старше 2017 года) "1" Августа 2018 г. | Размер статьи (от 400 слов) 2059 |
| Прямая полная ссылка на источник и сокращённая ссылка (bit.ly, goo.gl, tr.im и т.п.) https://habr.com/company/ruvds/blog/418823/ , сокращенная ссылка: https://bit.ly/2ymxAbB | | |
| Теги, ключевые слова или словосочетания Python, разработка | | |
| Перечень фактов, упомянутых в статье <ol style="list-style-type: none">1. Есть 3 возможные причины: 1) GIL 2) интерпретируемость языка 3) динамическая типизация.2. Интерпретатор может выполнять лишь одну операцию за раз, независимо от того, как много потоков имеется в программе. (GIL).3. В PyPy есть GIL, он обычно более чем в 3 раза быстрее, чем CPython4. Большую часть времени (если только это не код, который запускается лишь один раз) Python занимается выполнением готового байт-кода.5. JIT не ускоряет выполнение кода, однако позволяет выполнять оптимизации кода в процессе его выполнения.6. JIT компилятор сравнительно долго запускается - возможная причина его отсутствия в CPython7. Проект Cython объединяет Python и статическую типизацию, что приводит к 84-кратному росту производительности в сравнении со стандартным Python.8. Причиной невысокой производительности Python является его динамическая природа и универсальность. | | |
| Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта) <ol style="list-style-type: none">1. Для оптимизации скорости работы приложений, время запуска которых неважно, а производительность которых может выиграть от использования JIT-компилятора, стоит использовать PyPy.2. Если нужна максимальная производительность и ограниченность статической типизации не важна — стоит использовать Cython.3. Благодаря своей гибкости, Python удобен в создании прототипов проектов. | | |
| Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта) <ol style="list-style-type: none">1. Как правило, при выборе Python, проблема производительности не должна волновать.2. При выборе языка разработки почти всегда придется чем-то жертвовать: либо скоростью исполнения программы, либо скоростью разработки.3. Однозначной причины медлительности Python не найдено. | | |
| Ваши замечания, пожелания преподавателю или анекдот о программистах¹ | | |

¹ Наличие этой графы не влияет на оценку