



Язык программирования Julia

Швачко Никита

Дмитрий Перцев

Язык программирования Julia

- Julia — высокоуровневый, динамически типизируемый язык, рассчитанный на научные и числовые вычисления, с целью обеспечить близкую к C/Fortran скорость при удобстве Python/R-подобного синтаксиса
- Базовая библиотека и стандартные типы встроены в язык и доступны без внешних зависимостей, что ускоряет разработку и прототипирование
- Богатая экосистема пакетов: линейная алгебра, оптимизация, статистика, визуализация, работа с данными и др.
- Активное сообщество и растущая популярность в научных и инженерных кругах

Области применения Julia

- Julia изначально создавалась как язык для высокопроизводительных численных и научных расчетов. Его активно используют в физике, математике, инженерии, биоинформатике, климатическом моделировании, астрономии и других фундаментальных и прикладных науках. Для этого существуют специализированные пакеты для решения дифференциальных уравнений, оптимизации, работы с матрицами и др. Julia широко применяется для сложных симуляций, научных моделей и обработки больших научных данных.

Причины создания

- Создатели Julia называют себя «жадными» программистами, потому что хотели объединить лучшие качества множества языков — C, Python, Lisp, Matlab, R и других.
- Они стремились создать язык, который был бы одновременно быстрым, как C, динамичным, как Ruby, и удобным для научных вычислений, как Matlab и R. Julia задумывалась как открытый, компилируемый язык с возможностью интерактивной работы, простым синтаксисом и мощными средствами, такими как макросы и множественная диспетчеризация. Авторы хотели избавиться от громоздкости систем вроде Hadoop, сохранив их распределённую мощь.
- Цель состояла в том, чтобы один язык подходил для всего — от простых скалярных операций до масштабных параллельных вычислений. Несмотря на амбициозность задачи, они реализовали уже 90% своих идей и приглашают других «жадных» программистов присоединиться к развитию Julia.

История языка Julia

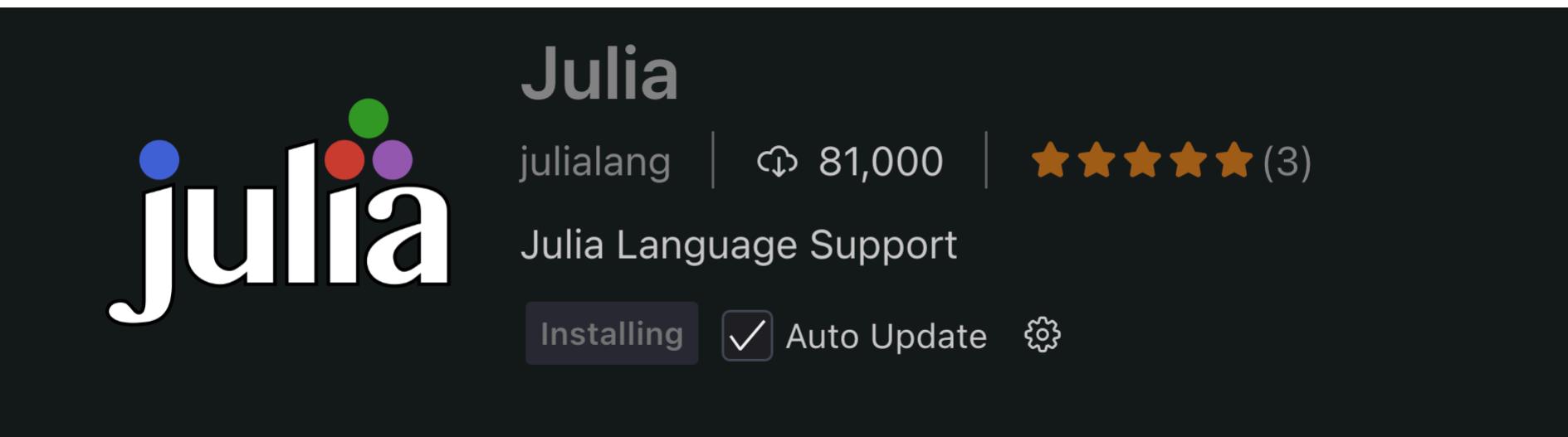
- 2009: работа над Julia началась группой учёных МИТ (Джефф Безансон, Стефан Карпински, Вирал Шах, Алан Эдельман). Целью было совместить удобство высокоуровневого языка и скорость C/Fortran.
- 2012: опубликована первая открытая версия языка (февраль 2012) . В апреле 2012 был запущен сайт проекта с описанием целей языка.
- 2018: выпущена версия 1.0, после чего синтаксис Julia стал стабильным и была обещана обратная совместимость. Это означало, что написанные на Julia программы останутся работоспособными при обновлениях языка.
- 2014–настоящее время: ежегодно проводится конференция JuliaCon, популярность языка растёт.
- В 2019 году создатели Julia получили престижную премию Wilkinson Prize за вклад в разработку высокопроизводительного научного ПО .

Основные идеи Julia при создании

- Быстрая: Julia компилируется в эффективный машинный код через LLVM.
- Универсальная: Множественная диспетчеризация упрощает ООП и функциональные паттерны; есть богатая стандартная библиотека.
- Динамическая: Динамически типизирована и удобна для интерактивной работы.
- Техническая: Оптимальна для численных вычислений, с удобным синтаксисом и встроенным параллелизмом.
- С optional-тиปизацией: Типы можно указывать для ясности и надёжности кода.
- Композиционная: Пакеты легко совмещаются и сохраняют высокую производительность.

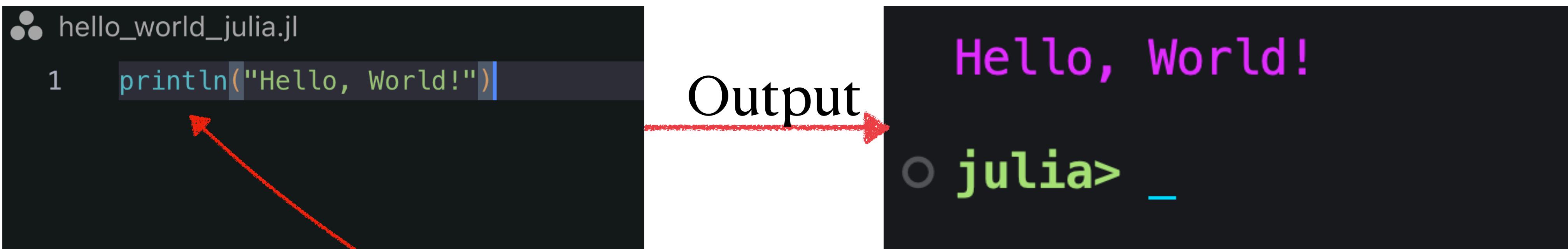
Способы запуска Julia

- **Онлайн компилятор:**
 - <https://coderunbox.com/playground/playcode-julia?active=script.jl>
 - <https://www.tutorialspoint.com/compilers/online-julia-compiler.htm>
- Способ запуска через VScode:
 - Установка языка по инструкции с официального сайта: <https://julialang.org/install/>
 - Установка расширения в vscode



- Готово

Программа "Здравствуй, мир!" с объяснением



- выводит текст (или любое значение) в стандартный поток вывода
- и добавляет перевод строки (`\n`) в конце.

Конец

- Список литературы

julialang.org - оф. сайт языка

[https://ru.wikipedia.org/wiki/Julia_\(язык_программирования\)](https://ru.wikipedia.org/wiki/Julia_(язык_программирования)) - википедия

План следующих 5 презентаций

1. Данные и переменные в Julia
2. Конструкции потока управления
3. Функции и обработка ошибок
4. Объектно-ориентированное программирование
5. Конкурентность / параллелизм