

Ruby — Lurkmore

Ruby (рус. Рубин, транс. Руби, униз. Раби) — интерпретируемый (скриптовый) [язык программирования](#) общего назначения, а также DSL для Rails.

История

Некий [расовый](#) японский [программист](#) [Юкиhiro Мацумото](#) aka Matz, несмотря на более чем десятилетний опыт разработки, так и не смог осилить ни один из скриптовых языков и, сославшись на их ущербность, создал свой собственный язык, состоящий почти полностью из ништяков, спиженных из других популярных языков, чего и не скрывает. При этом на данный момент языку уже 20 лет, а известен он стал буквально [последние несколько лет](#), и то благодаря написанному на нем фреймворку Rails и наконец-то доведенной до ума документации.



Как перл, только красный

Плюсы и минусы

За что любят

- ООП, причем в стиле [SmallTalk](#), а не [крестов](#)
- Активное комьюнити
- Няшный синтаксис
- Высокая зарплата
- Поддержка кучи модных фиш
- Rails
- Гемы, тысячи их (хотя по сравнению с другими языками, либ в рубях таки маловато)
- Можно пилить либы на [C](#), что выводит производительность приложения на космический уровень (если [не наговнокодить](#))
- Поднимает [ЧСВ](#) пишущих на нем

За что ненавидят

- Плохо работает на венде
- Отсутствие реальной многопоточности и этот ваш GIL [\[1\]](#)
- Очень мало вакансий и 95% из них на рельсы [\[2\]](#)
- Родной интерпретатор MRI достаточно медленный, и умеет использовать только одно ядро процессора
- Жрет памяти, как не в себя
- Проект, написанный ранее на руби и получивший должное развитие, как правило, переписывается на [жабу](#) или скалу [\[3\]](#)
- Нету нормальных доков на русском
- Высокий порог входа
- Минималистичная стандартная библиотека, из-за чего на любой чих приходится доставлять сторонние либы, каждая из которых, в свою очередь, зависит еще от десятка-двух, что приводит к настоящему аду зависимостей в проекте
- Стандартная библиотека, а вслед за ней и все остальные жутко любят менять API, что ломает не только обратную, но и прямую совместимость, причем в минорных версиях — проект, написанный на 2.1.5, вполне может не запуститься на 2.2.0
- Одиночное наследование без абстрактных классов и интерфейсов порою может вылезти боком
- Некоторые [новоиспеченные адепты](#) настолько увлекаются переопределением всего и вся и конструированием классов на лету, что порождают тонны абсолютно нечитаемого кода в лучших традициях [Perl'a](#)
- [Хипстерский](#)

Чем доставляет

Руби ценится в первую очередь за скорость [разработки](#), [синтаксис](#), ориентированный на [рутину](#), кучу [готовых](#) решений, а так же уникальное сочетание поддержки всякой функциональной [модноты](#) и традиционного императивного [подхода](#). Встроенная в язык поддержка профайлеров и тестов заставляет здорово оргазмировать тех, кто знает, что это такое, и просто повышает ЧСВ у остальных. При этом степень задротства регулируется в зависимости от задач, а наличие хорошей документации на английском, а иногда и на русском, снижает порог входа до [минимума](#). Скорость работы скриптов на руби сливается аналогичным пистоновским всего в 7 из 10 [тестов](#), что вполне вменяемый результат для [интерпрайзного](#) языка. Ну а еще говорящие на [C](#) могут высказаться [тут](#), и если повезет, то стать

соавтором языка.

Имплементации

Существует несколько вариантов интерпретаторов языка ruby, основным считается MRI, который пилят Matz.

- MRI — Matz ruby implementation, является тем самым православным и расово верным руби интерпретатором японца
- JRuby — Интерпретатор, выполняющий ruby код в JVM машине, за счет чего получает реальную многопоточность и возможность юзать ништяки из [жабы](#). Используется, впрочем, редко, так как во-первых, теряется возможность юзать библиотеки, напиленные с расчетом на MRI, а во-вторых, по отжиру памяти и тормознутости является бесспорным лидером среди большого семейства интерпретаторов, использующих JVM
- IronRuby — то же, что и JRuby, только под [.Net](#)
- Rubinius — уже давно грозит показаться все другим имплементациям кузькину мать, но на данный момент так и не взлетел, не смотря на большое кол-во инвестиций.

Rails

Веб-фреймворк некоей компании 37signals Basecamp, [внезапно](#) вызвавшая интерес, как и к самому фреймворку, так и в целом к языку ruby. Бытует мнение, что, если б не рельсы, вполне вероятно, что о японской поделке никто бы и не узнал. На данный момент по праву считается одним из самых развитых веб-фреймворков, однако пальма первенства уже давно и успешно [проебана](#): та же [питоновская](#) Джанга развивается в разы быстрее и обрастает фичами чуть ли не каждый день, да и всякие [похапешные](#) поделки типа Symfony и Yii и даже серверный [жабоскрипт](#) в лице Node.js уже обогнали Рельсы по популярности [\[4\]](#).

Рельсы любят:

- За простоту и удобство
- Скорость разработки
- Не надо знать этот ваш [SQL](#), ведь есть удобный ActiveRecord
- Даже jquery ставится с помощью rubygems
- Поддержка таких [ништяков](#) как CoffeeScript, SASS, LESS из коробки

Рельсы хейтят:

- Рельсы решето
- Для масштабирования проекта на рельсах требуется вливание [тонн нефти](#), алсо это проблема скорее интерпретатора MRI, чем веб-фреймворка и языка в целом
- Хоть сколько-нибудь серьезную работу с базой ActiveRecord не вывозит, и [SQL](#) учить таки приходится
- Кроме того ActiveRecord сам по себе демонстрирует не самый удачный подход к построению ORM, и при достаточно сложной структуре базы даже при условии не сильно большой нагрузки работать становится неудобно
- Разработка под windows = боль

Рельсы — решето

Локальный мем в ruby комьюнити, носит схожий смысл с [не падающей плазмой](#). Долгое время, рельсоводов не заботила ситуация с безопасностью: считалось, что уязвимости — удел [пехапэ](#) со всякими cms by Vasya Orasniy, а их это не касается. [Внезапно](#) это оказалось не так.

Истоки

Все началось, с того, что [оянн школьник](#) из [этой страны](#) Егор Хомяков обнаружил уязвимость, и отправил репорт в багтрекер rails на гитхабе. Но был послан контрибьютарами, после чего разгневанный школьник решил устроить [экстерминатус](#) и вообще показать кузькину мать. И таки сделал это, используя ту же самую уязвимость в гитхабе, который в качестве веб-фреймворка использует... да, анон, таки рельсы. После чего Егор смог отправить коммит в репозиторий rails, несмотря на то, что как таковых прав на данное действие он не имел. От этого и гитхаб, и контрибьюторы рельсов начали [срать кирпичами](#), [школьника](#) на время забанили, а багу таки исправили.

Продолжение банкета

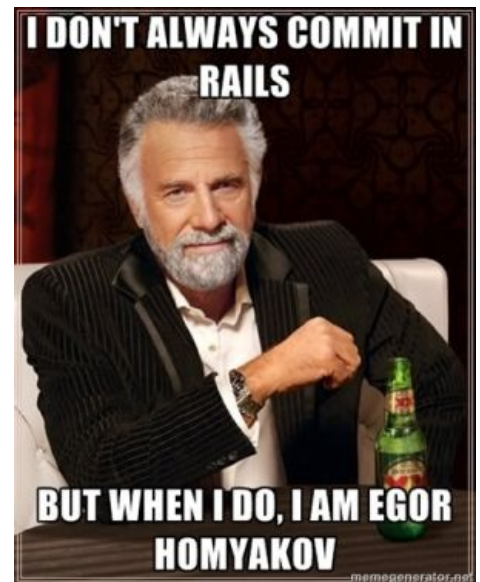
В целом весь последующий 2013 год, параллельно с 4 версией

рельсов, пилились периодические секьюрити фиксы. А очередное обновление рельсов всегда сопровождалось фразами: решето, друшлак, рельсы решето, etc...

Холивары

Ruby vs Python

Так как оба языка довольно похожи, то периодически какой-нибудь пэхапэшник, уставший от постоянных унижений со стороны программистов, решает перейти на что-то крутое и, естественно, задает вопрос на какой-нибудь [хабре](#) [5], [6]. В качестве аргументов пистончики традиционно тыкают в свою интерпрайзность, серьезность, четкий code-style и производительность (весьма относительную, но в сравнении с рубями таки нехилую). На что рубисты отвечают, что джанга хуйня, а рельсы рулят, и не забывают про давнюю боль пистона со второй и третьей версией, от которой питоноводы так и не избавились (справедливости ради, в рубях при переходе даже между минорными версиями можно огрести достаточно похоже проблемы). Потом обычно идут унылые обсуждения синтаксиса, отступов питона и библиотек. Стоит отметить, что в плане web-разработки большой разницы между этими двумя языками нет, а ни для чего, кроме этого, Ruby, в отличие от Python'a, практически не используется.



Rails != Ruby

«Ruby? Это такой DSL для написания Rails-приложений?»

— lor



Тентакля

Несмотря на всю крутость раби как языка, не существует ни одной вакансии отличной от рельс (иногда может идти вместе с синатрой). При этом в качестве значимости фреймворка обычно указывается, что когда-то [твиттер](#) работал на сильно переписанных под себя рельсах, правда есть еще и [гитхаб](#). Проектов же, написанных на самом раби, в природе достаточно мало, даже на [cyberforum](#) он лежит в разделе веб-языков, хотя это язык общего назначения, что символизирует и отлично троллит начинающих рельсоводов. Сами же рельсы ни чуть не менее няшные и удобные, чем раби, задают тренды среди веб-фреймворков и охуеть как тормозят даже на очень мощном железе, что лечится кэшированием всего и вся либо же переходом на [синатру](#).

Ruby и Windows

Суший ад и погибель ждут любого, кто попытается разрабатывать на этом языке под вендой. Суть в том, что Ruby разрабатывался как язык максимально открытый, и родная платформа для него Linux. Большинство гемов (местное название библиотек) просто не тестируется под вендой. Инсталлятор языка выглядит как убогий костыль, содержит в себе вперемешку исходники интерпретатора, компилятор Си и еще какие-то хаки, которые не работают, если, например, в пути к папке установки есть кириллица или пробелы (причем ошибка выяснится только при попытке установить какой-нибудь гем, написанный на Си и требующий нормального компилятора). А, например, драйвер для Mysql вообще не работает, пока не перекинешь dll из папки мускула в папку с гемами. Можно, конечно, использовать SQLite, но кому он нахуй нужен. А также отправка почты, настройка веб-сервера, отличного от стандартного, и еще тонна всяких проблем, о которых лучше не знать. Поэтому грамотный анон заранее ставит виртуальную машину с [бубунтой](#) и наслаждается разработкой под рельсы, ну или качает готовый образ с настроенным окружением [7].

Что в итоге

В реальности же руби вполне вменяемый язык с высокой зарплатой, почти как у жабы и шарпа, при этом в разы их няшнее и продуманнее, с кучей синтаксического сахара и годным, хоть и относительно небольшим комьюнити. Но, несмотря на простоту и юзерфрендливость, сделать что-то годное без полного понимания, что ты делаешь не получится: нужно будет читать и вникать в [манулы на английском](#). Свой вклад вносит коварный Python — порог входа ниже, комьюнити сильно больше, как следствие предыдущего — наличие [сотен библиотек и фреймворков](#) на все случаи жизни, при этом область применения не ограничивается одним только вебом, а по количеству вакансий одна только Джанга сопоставима со всеми рубями. И все, чем придется пожертвовать за эту радость — это отсутствие некоторого синтаксического сахара, что на самом деле заставляет писать более однообразный и, соответственно, понятный код. Profit жи. И самое главное, почему начинающий быдлокодер не любит руби — это то, что поднять визитку и правда быстрее и легче на пыхе, а объяснить заказчику, почему первокурсник Вася может сделать сайт за 5к рублей, а ты за 50к, очень сложно. При этом, когда твой уровень и уровень твоих заказчиков поднимается до такого, что за указание пыха в твоём резюме будут

громко смеяться, ты уже вряд ли захочешь вообще программировать, и скорее всего будешь каким-нибудь анал-итиком или менеджером проекта, которому уже глубоко похуй, на чем там пишут его прогеры, да и зарплата там уже совсем другая. С другой стороны, если ты [джуниор](#) и готов учиться, то руби научит писать более-менее красивый код. Ну или не научит ничему.

Примеры использования

Хеллоу ворд

```
puts 'Hello World' # Обычный вариант, это, кстати, комментарий

p 'Hello World' # Короткий, перлота

printf("%d",10); # Чистый Си
```

Синглтон

На жабе

```
public class Singleton {
    private Singleton() {}

    private static class SingletonHolder {
        public static final Singleton instance = new Singleton();
    }

    public static Singleton getInstance() {
        return SingletonHolder.instance;
    }
}
```

На руби, прогресс налицо

```
class JavaFucker
  include Singleton
end
```

Циклы, [moar](#)

```
10.times { p 'ima loop yo' }

for i in 0..10
  p "another loop #{i}"
end

#это не цикл! это замыкание (closure)
#а то что оно похоже на цикл и есть его главная синтаксическая прелесть
#в отличие от богомерзких сотен вложенных фунуций как в этом вашем JavaScript'e
#yield, господа!

(0..10).each do |i|
  puts "Python sucks #{i} times"
end
```

Добавление методов к существующему объекту

Способ 1

```
obj = SomeObject.new # Создание и инициализация объекта

def obj.new_method # Добавление нужных методов
  "do some things"
end
```

Способ 2

```
module AdditionalMethods # Создание модуля с нужными методами
  def new_method
    "do some things"
  end
end

obj = SomeObject.new
obj.extend(AdditionalMethods) # Пристыковали методы из AdditionalMethods к одному объекту
```

Интересные факты

- Хотя автор языка и против написание гуев на руби, клепать формочки можно на [QT](#) и [WxWidget](#)
- На руби можно писать игры [Gosu](#)
- ВНЕЗАПНО является скриптовым языком плагинов для [w:RPG Maker](#) по VX ACE включительно и

вполне успешно работает там под виндой. Более поздние версии (начиная с MV) таки сдались под напором [Web 2.0](#) и используют [JavaScript](#)

- Только False и nil возвращают False, 0 == True, в отличие от Жабы, Цэ и вообще практически всех мейнстримовых языков, так что зачастую доставляет примерно такой же разрыв шаблона, как нумерация массивов с единицы в [Pascal](#).
- Для руби есть интерпретатор, [написанный на руби](#)
- [Жаргон](#)

Ссылки

- [Официальный сайт языка, теперь и на русском](#)
- [67 000 дополнений!](#)
- [Ruby Warrior — интерактивный учебник](#)
- [Довольно годные статьи по Ruby на русском](#). На других страницах много инфы по RoR и прочему.

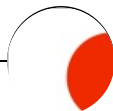
См. также

- [Хипстер](#)
- [perl](#)
- RUBY - англоязычный [вакланастер](#) на четвёртый движок, ещё до своего выхода в свет успевший повидать скандал по поводу своей [расово](#) латиноамериканской мордашки^[8]



Языки программирования

++i + ++i 1C AJAX BrainFuck C Sharp C++ Dummy mode Erlang Forth FUBAR
God is real, unless explicitly declared as integer GOTO Haskell Ifconfig Java JavaScript LISP
My other car Oracle Pascal Perl PHP Prolog Pure C Python RegExp Reverse Engineering
Ruby SAP SICP Tcl TeX Xyzy Anti-паттерн Ассемблер Быдлокодер
Выстрелить себе в ногу Грязный хак Дискета ЕГГОГ Индусский код Инжалид дежице
Капча КОИ-8 Костыль Лог Метод научного тыка Очередь Помолясь Проблема 2000
Программист Процент эс Рекурсия Свистелки и перделки Спортивное программирование
СУБД Тестировщик Умение разбираться в чужом коде Фаза Луны Фортран Хакер
Языки программирования



Япония

2channel AMV Dance Dance Revolution Futaba Geddan Guilty Gear Hard Gay Hello Kitty
Imaichi-tan IOSYS J-Rock Ja MAD Magibon MSX OS-tan Paper Child Ruby Sage
Silent Hill Sonic the Hedgehog Special Feeling Subaru Super Sentai Touhou Project Yaranaika
Zoomjar Айкидо Аниме Аокигахара Аум Синрикё Бичпакет Буккакэ Вап
Гайдзин ёнкома Годзилла Гуро Десу Дзен Длиннокот Дорама Иероглиф
Имдинская война Именные суффиксы Имиджборд Кавай Камикадзе Канакапча Караоке
Катана Ковай Консоли Косплей Курильские острова Кэри-Хироюки Тагава Лоли
Лэйдзи Мацумото Макото Синкай Манга Мику Хацуне Митрополит Токийский Даниил
Монорельс Моэ-антропоморфизм Мунспик Нанодесу Невада-тян Нека Некомими
Ниндзя Ня Обыкновенный японский школьник Отаку Педобир Поделки из бумаги
Реклама/Японская реклама Рулесрач Русско-японская война Самурай Сатоси Кон
Синтоизм Система Поливанова Советско-японские войны Суши Схватка двух йокодзун
Тамагочи Танака-нэко Тиёми Хасигути Три обезьяны Фингербокс Флаг на Иводзиме Фугу
Фукусима Хагакурэ Хаяо Миядзаки Хикки Хироо Онода Эроге Япония Японский идол

w:Ruby en:w:Ruby (programming language) ae:Ruby