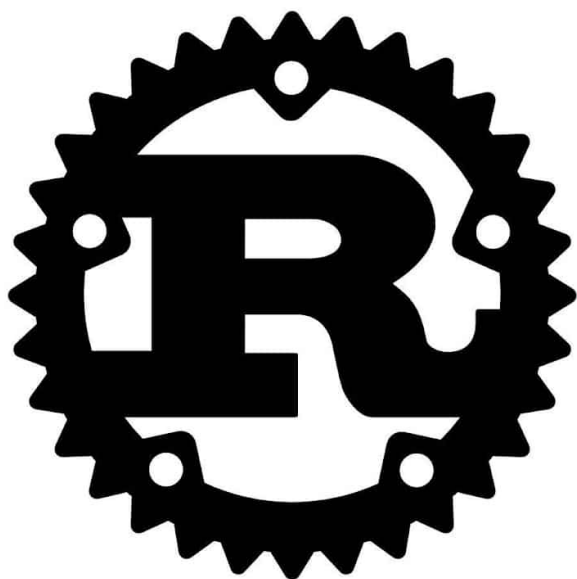


Rust против Python: какой из них лучше для вашего проекта?

Описание

Rust против Python – популярная дискуссия в наши дни. Хотя оба этих языка можно использовать для программирования общего назначения, один лучше другого в определенных сценариях и нишах. Чтобы знать, какой из них следует использовать для вашего проекта, вы должны иметь очень хорошее представление о функциях и архитектуре обоих. Эта статья Rust против Python поможет вам в этом.

Что такое Rust?



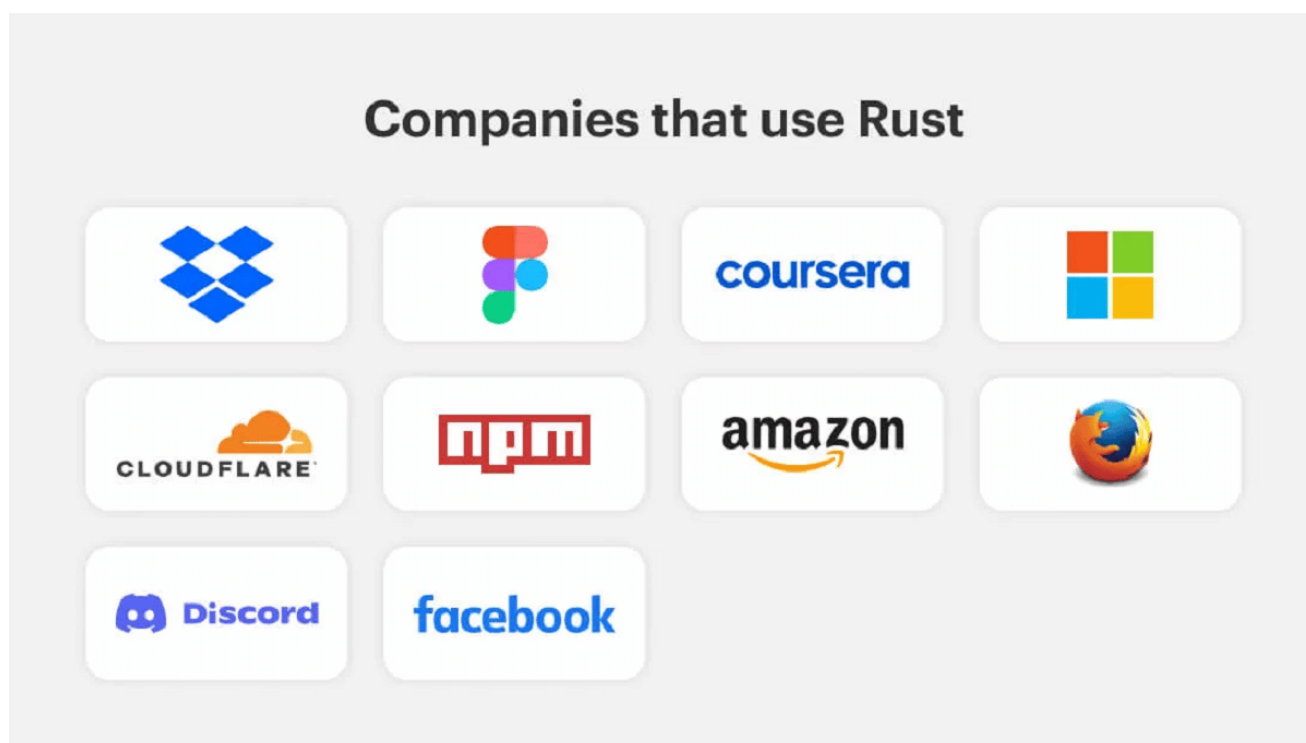
The Rust Programming Language

Логотип языка программирования Rust. (Источник изображения:
[официальный сайт Rust](#))

)

Rust – это относительно новый [язык программирования](#), который быстро стал популярным благодаря своей способности создавать высокопроизводительные, экономные по объему памяти и надежные приложения. Этот статически типизированный язык программирования имеет синтаксис, аналогичный C ++. У него нет никакой сборки мусора или времени выполнения. Следовательно, Rust предоставляет решения многих проблем в C ++, включая проблемы параллелизма и управления памятью. Несмотря на то, что это относительно молодой язык, Rust занял 7-й год как самая любимая технология в [опросе разработчиков StackOverflow 2022](#).

Кто использует Rust?



Компании, использующие Rust. (Источник изображения: [OpenXcell](#))

С тех пор, как Rust был представлен миру, многие компании, от стартапов до крупных организаций, начали внедрять его для улучшения существующих и создания новых программных приложений и сервисов. Сейчас многие ведущие мировые технологические компании используют Rust. Вот как они его используют. Посмотрите, как Kinsta противостоит конкурентам.

1. **Amazon:** Amazon использует Rust для своих [облачных сервисов](#), таких как

Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage [Service \(S3\)](#) и CloudFront. Amazon также является одной из компаний, которые вносят свой вклад в Rust.

2. **Dropbox:** Dropbox, ведущий мировой поставщик файлового хостинга, использует Rust в своей службе синхронизации файлов, которая является одним из его важнейших компонентов.
3. **Google:** Google – еще одна компания, входящая в состав Rust foundation. Он использует Rust для разработки Android и является основным языком при разработке операционной системы на ядре Linux.
4. **Microsoft:** Microsoft использует Rust в своей операционной системе Windows, и они выпустят новую версию Rust, которая позволяет разработчикам использовать Windows API для разработки программного обеспечения.
5. **Coursera:** Одна из ведущих мировых платформ онлайн-обучения использует Rust для выполнения заданий по программированию.

Что такое Python?



Логотип Python.

Python – один из самых [популярных языков сценариев](#). Python используется во многих областях, включая машинное обучение, науку о [данных, визуализацию](#) данных, анализ данных, DevOps, автоматизацию и тестирование. Простой синтаксис Python делает его идеальным выбором для начинающих разработчиков по всему миру. [Опрос разработчиков Stack Overflow 2022](#) года показал, что это третий по популярности язык среди профессионалов и людей, обучающихся

программированию. Python отдает предпочтение [объектно-ориентированному программированию](#), хотя вы можете использовать его и для функционального программирования.

Кто использует Python?



Популярные компании, использующие Python. (Источник изображения: [DailySmarty](#))

Python широко используется в ИТ, образовании, правительстве и финансовых отраслях. Ниже приведены ведущие технологические компании, использующие Python.

1. **Facebook:** Инженеры Facebook используют [Python наряду с PHP](#), потому что Python позволяет им быстрее разрабатывать и развертывать приложения и функции. Python используется в управлении инфраструктурой Facebook и множеством сервисов, включая Facebook Ads API и платформу Async IRCBot на Python.
2. **Google:** Google использует Python в своих сценариях из-за его простоты и эффективности. Google заменил свой первый паук для сканирования веб-страниц на Java на Python для [очистки веб-страниц](#).
3. **Netflix:** Netflix – это компания, которая использует Python для анализа данных. Кроме того, они используют его в своем центральном шлюзе оповещений, который отправляет оповещения пользователей, и в приложениях monkey, которые отслеживают изменения в безопасности.
4. **Spotify:** Spotify – еще одна компания, которая использует Python для анализа

данных. Есть много других мест, где они используют Python. Они используют [фреймворк Django](#) и сетевую библиотеку ZeroMQ для [внутренних](#) приложений. Кроме того, они используют его для создания процессов, сценариев и прототипов.

5. **Instagram:** Instagram, приложение для социальных сетей, которое позволяет обмениваться изображениями и видео, расширило свою инфраструктуру с помощью Python.

Ключевые особенности Rust

Rust объединяет лучшее из C ++ и дополнительные функции. Четыре основные ключевые особенности Rust – это безопасность памяти, потокобезопасность, кроссплатформенная поддержка и абстракции с нулевыми затратами.

Безопасность памяти

При использовании такого языка, как C ++, многие проблемы с памятью могут снизить производительность приложения. Rust решает большинство проблем с управлением памятью с помощью своей 'модели владения' и системы типов, позволяя программам анализировать память во время компиляции, а не во время выполнения. Это означает, что переменные владеют своей памятью и могут заимствовать память у других переменных. Эта функция устраняет необходимость в сборщике мусора и гарантирует, что программы Rust не содержат ошибок во время компиляции.

Безопасность потоков

[Потокобезопасность](#) – это тип системы, который гарантирует отсутствие условий гонки во время выполнения. Когда два или более потоков одновременно обращаются к одной и той же ячейке памяти, возникает состояние гонки, которое может привести к неправильному поведению программы. С моделью владения данными Rust потоки могут записывать в переменную только в том случае, если они владеют ею или изменяют ее. Это гарантирует, что только один поток может получить доступ к переменной одновременно.

Абстракции с нулевой стоимостью

По мере продолжения разработки вам может потребоваться использовать больше абстракций в коде, чтобы улучшить обслуживание кода и упростить добавление дополнительных функций. Использование большего количества абстракций в коде может повлиять на производительность ваших программ во время выполнения. Rust гарантирует, что абстракции будут стоить ноль. Это означает, что абстракции не требуют затрат времени выполнения, или код с абстракциями похож на код, в котором нет никаких абстракций.

Кросс-платформенная поддержка

Rust позволяет разрабатывать приложения на [нескольких платформах](#), таких как Windows, [Linux](#), macOS и т.д. Rust может легко обрабатывать конфигурации на разных платформах. Кроме того, его инструмент сборки cargo помогает развернуть код на нескольких платформах с помощью одной команды.

Ключевые особенности Python

Python изначально разрабатывался для удобства чтения кода. Поэтому его синтаксис близок к английскому языку. Кроме того, Python используется различными академическими группами для математических и научных работ. Поэтому он имеет большой набор библиотек для математики и многочисленных вычислений.

Простой синтаксис, подобный английскому

[Python](#) – один из самых любимых языков программирования общего назначения среди людей, склоняющихся к кодированию, из-за простого синтаксиса, похожего на английский, который упрощает написание кода. Поэтому разработчики-любители находят обучение легким, и многие используют его в качестве начального языка программирования. Кроме того, его простая синтаксическая структура позволяет ускорить разработку. Вы можете написать программу на Python с меньшим количеством строк кода, чем на других языках. Это в конечном итоге сокращает время развертывания приложения на рынке.

Динамически типизированный

Вам не нужно определять типы переменных в Python, потому что интерпретатор

Python выполняет эту работу за вас во время выполнения программы на основе выражения переменной. Таким образом, вы можете сосредоточиться непосредственно на функциональности, а не тратить время на правильное определение типов переменных.

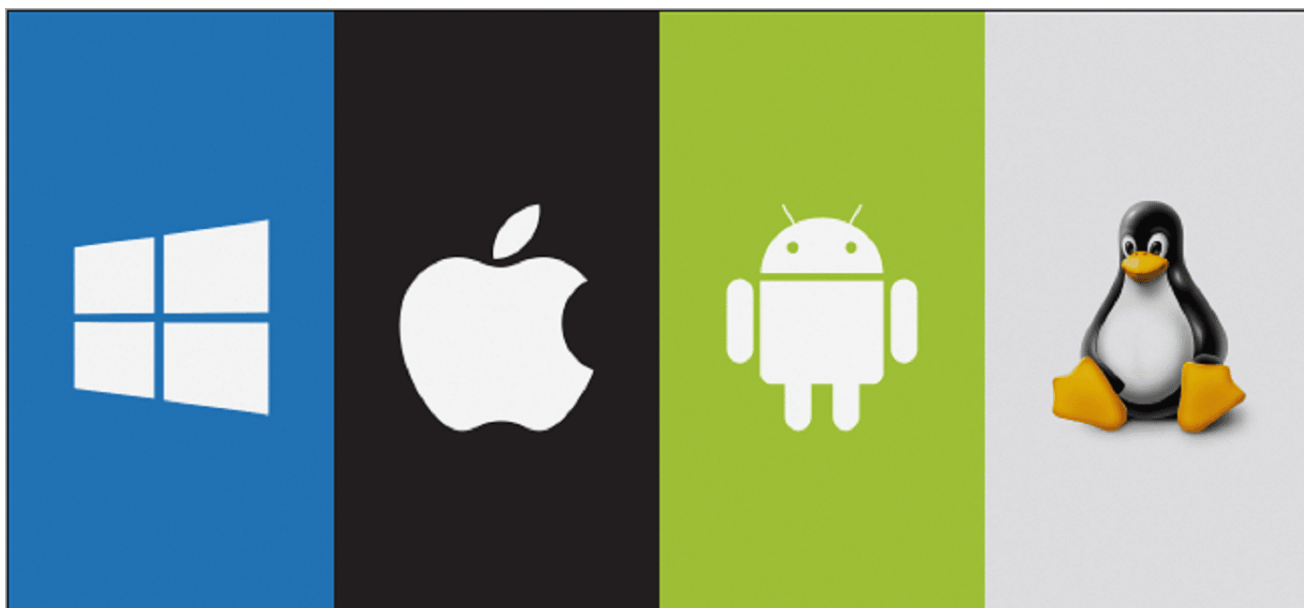
Большое сообщество

У Python есть большое сообщество с миллионами людей по всему миру, которые поддерживают и вносят свой вклад в его развитие. Это большое сообщество является преимуществом для любого разработчика Python, потому что они могут быстро найти решения любых проблем, с которыми сталкиваются при разработке. Кроме того, это означает, что у вас есть множество [ресурсов для изучения Python](#), таких как документация, руководства для разработчиков, видеоролики и учебные пособия.

Большая стандартная библиотека

Python; стандартная библиотека состоит из множества фреймворков, модулей и библиотек с многократно используемым кодом для разработки практически всего, что вы хотите. Таким образом, вам не придется тратить время и силы на разработку всего с нуля.

Кросс-платформенная поддержка



Поддержка Python на нескольких платформах. (Источник изображения: [TheGameRoof](#))

)

Python позволяет запускать приложения на [нескольких платформах](#), таких как Windows, Linux, macOS и т.д., устраняя необходимость написания отдельного кода для каждой платформы.

Плюсы и минусы использования Rust

Rust имеет много преимуществ по сравнению с недостатками. Давайте выясним, что это такое.

Преимущества

Основным преимуществом Rust является его высокая производительность. Но это не единственный. Следующий список дает вам все основные плюсы Rust.

- Rust идеально подходит для приложений, критически важных для производительности, потому что Rust экономит память без сбора мусора или времени выполнения.
- Rust безопасен для памяти и потоков из-за своей модели владения и системы типов.
- Rust – это статически и строго типизированный язык, гарантирующий отсутствие ошибок в коде, связанных с неправильно объявленными переменными.
- Низкие накладные расходы Rust делают его идеальным для разработки встроенных программ
- [Сообщество Rust](#) помогает вам обсуждать проблемы и идеи с другими, организовывать мероприятия, конференции и т.д., а также предоставляет множество учебных ресурсов.
- Rust имеет богатую документацию.
- Он предоставляет несколько инструментов для повышения вашей производительности. Например, он включает в себя интегрированный менеджер пакетов и инструмент сборки, поддержку редактора с полезными функциями и компилятор с полезными сообщениями об ошибках.
- Вы можете найти инструменты командной строки, которые упрощают вашу работу с ящиками.экосистема ввода-вывода.
- Rust позволяет разрабатывать программное обеспечение с кроссплатформенной поддержкой.

Недостатки

У Rust также есть несколько недостатков, и вы должны понять их, прежде чем использовать язык. Некоторые из этих недостатков включают:

- Время компиляции Rust медленнее, чем у других языков. Он соответствует всему пакету или ящику во время компиляции. Таким образом, время компиляции может быть больше.
- Для его изучения требуется больше времени по сравнению с другими языками, такими как Python. Вы должны знать хотя бы один язык, чтобы эффективно его использовать. Это делает Rust не подходящим для людей, которые впервые учатся программировать.
- Rust не обеспечивает значительной поддержки исправления monkey.
- Он может создавать двоичные файлы большего размера.

Плюсы и минусы использования Python

Теперь, когда мы рассмотрели Rust, давайте подробнее рассмотрим преимущества и недостатки использования Python.

Преимущества

Многие разработчики любят Python, и легко понять, почему. Некоторые из основных преимуществ включают:

- Python – один из самых простых языков для изучения из-за его простого синтаксиса. Таким образом, он имеет меньшую кривую обучения и идеально подходит для разработчиков-любителей по сравнению со сложными языками, такими как Rust или [PHP](#).
- Python – это универсальный язык, используемый в нескольких вариантах использования, таких как веб-разработка данных, наука о данных, машинное обучение, создание сценариев, прототипирование, автоматизация тестирования и т. Д.
- Обширная стандартная библиотека Python позволяет разрабатывать все, что угодно, от простых численных вычислений до сложных проектов анализа данных с меньшим количеством кода.
- Динамический ввод текста экономит время на объявление типов переменных.
- В Интернете доступно множество ресурсов для изучения Python, таких как учебные [пособия, документация, руководства для разработчиков](#) и т. Д.
- Множество инструментов, поддерживающих разработку на Python, [таких как IDE, специфичные для Python, инструменты проверки кода](#) и т. Д. В Python также есть инструмент под названием GitPython для взаимодействия с

репозиториями git. [Изучение рабочего процесса Git очень ценно для веб-разработки.](#)

- Позволяет разрабатывать программное обеспечение с кросс-платформенной поддержкой.
- Поскольку Python – это интерпретируемый язык, который упрощает процесс отладки
- Поддержка большого сообщества поможет вам быстро получить поддержку по вашим вопросам и быстрее их решить.
- Python можно использовать с такими языками, как .NET и PHP и легко интегрируется с другими компонентами, такими как базы данных. Подождите? Вы думаете, [PHP мертв? Нет, это не так, по крайней мере, согласно статистике использования.](#) На самом деле, [разработчики PHP получают хорошую зарплату.](#)

Недостатки

До недавнего времени Python не был хорошим вариантом для [веб-разработки](#). Python был нацелен на другой домен. Хотя Python многое сделал для улучшения во всех аспектах, у Python все еще есть некоторые недостатки.

- Python не подходит для приложений, требующих высокой скорости, потому что Python является более медленным языком, чем другие языки, из-за его построчного выполнения и динамической типизации.
- Python потребляет большой объем памяти из-за своих функций, таких как гибкие типы данных. Таким образом, он не подходит для приложений с интенсивным использованием памяти.
- Динамическая типизация также может привести к errors во время выполнения, который не будет обнаружен во время компиляции.
- Он не подходит для мобильных приложений из-за низкой скорости и высокого использования памяти.
- Доступ к базе данных с использованием Python может стать затруднительным, поскольку он не так развит, как во многих других языках.

Rust против Python: сравнение лицом к лицу

Далее мы собираемся сравнить Rust и Python в различных аспектах, таких как возможности трудоустройства, функции, масштабируемость, расширяемость и

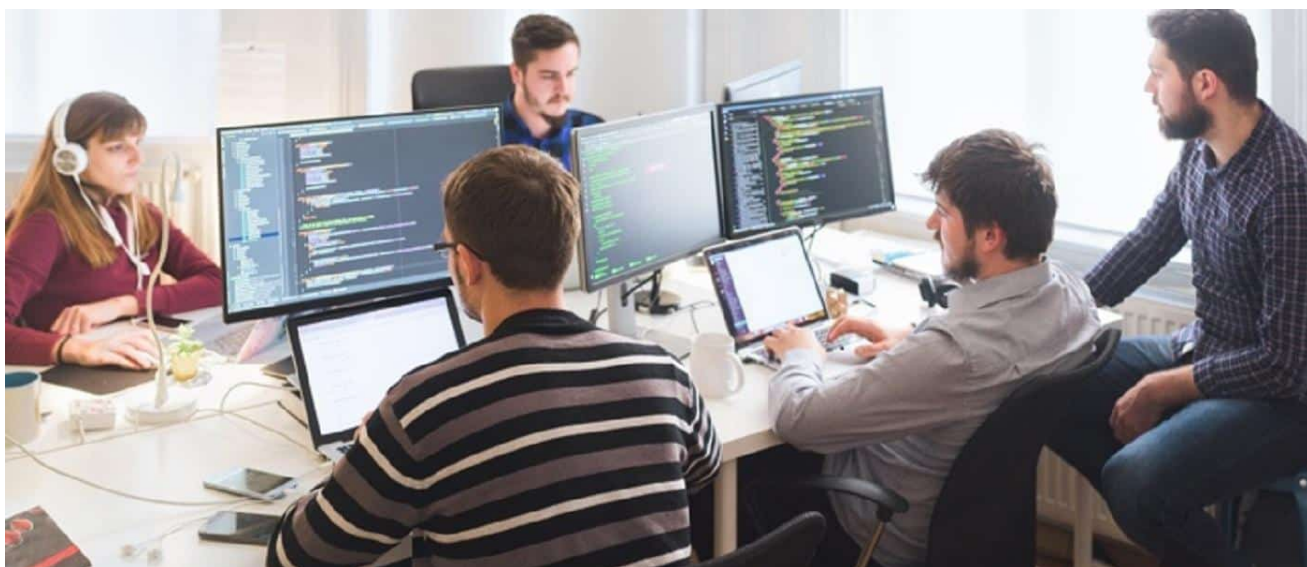
производительность.

Простота использования / обучения

Что касается простоты использования и обучения, Python опережает язык Rust. Как упоминалось ранее, Python стал одним из лучших языков программирования, используемых во всем мире, из-за его простоты в изучении. Если кто-то впервые [учится программировать](#), ему следует выбрать Python, а не Rust. Rust имеет более сложную синтаксическую структуру, чем Python. Следовательно, для его освоения требуются предварительные знания в области программирования.

Упрощенный опыт разработки Python и множество доступных модулей, библиотек и фреймворков упрощают его использование во многих областях. По этой причине Python используется в самых разных областях по сравнению с любым другим языком программирования. Исследователи считают его простым в использовании. Даже не имея никакого опыта программирования, они могут легко понять его концепции. Кроме того, они могут написать программу на Python, используя меньше строк кода, что потребовало бы намного больше строк кода для Rust. Таким образом, с точки зрения простоты использования и обучения Python является победителем.

Возможности трудоустройства



Возможности трудоустройства Rust против Python. (Источник изображения: [SEEK](#))

Python является незаменимым инструментом в стеках технологий для многих

отраслей промышленности по всему миру. Существует множество возможностей трудоустройства для квалифицированных программистов на Python. Благодаря широкому разнообразию вариантов использования, программисты на Python могут работать во многих областях. Например, разработчики Python могут работать в качестве помощников по анализу данных, машинному обучению и науке о данных или в качестве [разработчиков полного стека](#) с такими фреймворками, как Django и т. Д. Хотя JavaScript был специально создан для веб-разработки, [Python становится конкурентом Nodejs](#) даже в этой области.

По сравнению с Python, Rust – это новый язык. Но в последние годы использование Rust неуклонно росло. Кроме того, число разработчиков Rust намного меньше, чем число разработчиков Python в отрасли. Таким образом, в настоящее время существует более высокий спрос на программистов Rust в различных отраслях промышленности. На самом деле, [веб-разработчики Rust являются одними из самых высокооплачиваемых профессионалов](#). Итак, когда вы рассматриваете возможности трудоустройства, у программистов Python больше возможностей, чем у разработчиков Rust, но у Rust есть потенциал увеличить его в ближайшие годы.

Обработка ошибок

Rust и Python имеют разные механизмы обработки ошибок. Обработка ошибок Python не предоставляет много возможностей для улучшений во время компиляции. При возникновении ошибки Python выдает исключение во время выполнения, но не предлагает никаких предложений о том, как их исправить. В отличие от этого, Rust возвращает значение, когда обнаруживает ошибки во время компиляции и не останавливается. Он также предоставляет возможные исправления для него. Таким образом, обрабатывать ошибки в программах Rust проще, чем в Python. Кроме того, точное определение ошибок во время компиляции, а не во время выполнения, помогает создавать высококачественные программы Rust.

Расширяемость

Python по-прежнему опережает Rust из-за большого количества доступных библиотек, модулей, фреймворков и инструментов с точки зрения расширяемости. Благодаря этому вы можете использовать Python для написания простого скрипта для автоматизации повседневной деятельности и создания сложного веб-приложения без написания большого количества кода. Python

используется для многих целей в самых разных отраслях из-за его высокой расширяемости. По сравнению с Python экосистема Rust продолжает расти. Но в последние годы наблюдается быстрый рост. Реестр ящиков сообщества Rust продолжает добавлять новые ящики. Однако текущее количество ящиков не так расширяемо, как в Python.

Характеристики

Если учесть особенности каждого языка, Rust обладает более интуитивными и четкими функциями, чем Python. Как обсуждалось ранее, Rust безопаснее для памяти и потоков, чем Python. Программы Rust более эффективны, чем Python, без сбора мусора и времени выполнения. С другой стороны, Python имеет меньше отличительных особенностей, чем Rust. Примечательные языковые особенности Python включают динамическую типизацию, простой синтаксис и интерпретируемый язык общего назначения. Кроссплатформенная поддержка распространена на обоих языках. Поэтому, с точки зрения возможностей, мы можем сделать вывод, что у Rust больше возможностей, чем у Python.

Сборка мусора

У Rust нет сборки мусора во время выполнения, но у Python она есть. Поэтому Rust не проверяет, есть ли неиспользуемая память во время выполнения, что может замедлить выполнение программы. С другой стороны, Python проверяет наличие неиспользуемой памяти во время выполнения, что является одной из причин того, что программы на Python становятся медленнее, чем Rust.

Управление памятью

Python и Rust управляют памятью двумя разными способами. Python использует сборщик мусора для управления памятью, который всегда ищет неиспользуемые переменные в коде и освобождает их память. Напротив, Rust использует модель владения для управления памятью, которая проверяет любые нарушения правил владения во время компиляции. Кроме того, Rust использует стеки и кучи для хранения значений. В стеке хранятся скалярные значения, в то время как куча используется для нескалярных значений. Подход Rusts к управлению памятью более инновационный, чем Python, поскольку он управляет памятью более эффективно, чем обычный сборщик мусора во время выполнения.

Производительность

Если учесть аспекты производительности обоих языков, Rust является явным победителем. Rust предназначен для создания высокопроизводительных приложений. Как утверждает Rust, Rust “быстро развивается” благодаря своим неотъемлемым функциям, таким как отсутствие интерпретатора и эффективное управление памятью. С другой стороны, медлительность является одним из недостатков Python, что делает его непригодным для приложений с высокой производительностью. Низкая производительность Python включает в себя его интерпретатор и неэффективное управление памятью посредством сборки мусора во время выполнения.

Популярность

Если сравнить эти два языка с точки зрения популярности, Python – самый популярный язык программирования среди профессионалов и людей, которые учатся программировать. Постоянная популярность Python была выявлена благодаря ведущим в отрасли исследованиям использования программирования. Например, [опрос разработчиков Stack Overflow 2020](#) года показал, что Python является третьим по популярности языком в мире, а Rust даже близко к нему не приблизился. Интересно, что Python также смог сохранить свой статус популярности в последнем [опросе разработчиков Stack Overflow 2022](#) года. Однако результаты опроса показали, что Rust является самым любимым языком программирования, который он поддерживает уже несколько лет.

Масштабируемость

Такие особенности Python, как низкая производительность, неэффективное управление памятью и интерпретируемый язык, ограничивают его способность масштабировать приложение на Python для более широкой базы пользователей. Но ограничения масштабируемости Python можно преодолеть с помощью таких методов, как горизонтальное и вертикальное масштабирование базовой инфраструктуры.

Python используется многими технологическими компаниями с растущей базой пользователей. Таким образом, Python может сделать его масштабируемым при правильном использовании масштабирования инфраструктуры. С другой стороны,

Rust разработан для обеспечения высокой производительности. Таким образом, вы можете создавать масштабируемые веб-сервисы с помощью Rust без значительного масштабирования инфраструктуры и затрат.

Безопасность



Сравнение безопасности между Rust и Python. (Источник изображения: [TechRepublic](#))

Rust также предназначен для обеспечения большей безопасности, чем Python. В [руководстве по безопасности Rust](#) описывается, как он фокусируется на том, чтобы стать языком безопасности памяти. Все проблемы с памятью обнаруживаются во время компиляции, предотвращая несанкционированный доступ к переменным. Кроме того, команда Rust постоянно работает над устранением уязвимостей в системе безопасности экосистемы Rust. Вы можете просмотреть все исправления безопасности в их [списке рассылки объявлений о безопасности Rust](#). Напротив, Python не является безопасным для памяти и потоков до уровня Rust. Разработчикам необходимо управлять памятью и устранять утечки памяти.

Скорость

Скорость напрямую связана с объемом памяти и производительностью программы. Поскольку Rust более эффективен в отношении памяти и производительности, программы Rust имеют более высокую скорость выполнения,

чем программы на Python.

Поддержка (сообщество и документация)

В отличие от Rust, Python имеет обширное и разнообразное сообщество и документацию. Сообщество Python – это большое и активное сообщество, в котором вы можете внести свой вклад в разработку Python и легко найти ответы на любые вопросы. Он обладает обширной базой знаний, из которой вы можете узнать все о Python.

С другой стороны, Rust имеет относительно небольшое сообщество. Тем не менее, сообщество Rust быстро расширяется, а его документация хорошо написана и всеобъемлюща. Вы можете изучить Rust от А до Я.

Сравнительная справочная таблица для Rust против Python

В следующей таблице приведено краткое сравнение Rust и Python.

| Особенность | Python | Ржавчина |
|-----------------------------------|---|--|
| Простота использования / обучения | Простой в освоении и использовании благодаря простому синтаксису и обширным библиотекам | Сложный в освоении. Требуется предварительное знание программирования |
| Возможности трудоустройства | Более широкие возможности | Возможности ниже, но это один из языков, на котором можно получать более высокую базовую зарплату. |
| Обработка ошибок | Создает исключения во время выполнения. Нет рекомендаций по исправлению ошибок | Обработка ошибок во время компиляции с предложениями по исправлению ошибок |

| Особенность | Python | Ржавчина |
|--------------------|--|---|
| Расширяемость | Огромное количество библиотек делает его очень расширяемым | Низкая расширяемость по сравнению с Python |
| Характеристики | Не так много отличительных и интуитивно понятных функций | Более четкие и интуитивно понятные функции |
| Сборка мусора | Доступно | Не сбор мусора |
| Управление памятью | Через сборку мусора | Через систему типов и модель владения |
| Производительность | Низкая производительность | Высокопроизводительный |
| Популярность | Более высокая популярность, чем Rust Python | Низкая популярность, чем у Python |
| Масштабируемость | Менее масштабируемый, чем Rust | Высокая масштабируемость |
| Безопасность | Нет памяти или потокобезопасности | Более безопасный, чем Python, с памятью и безопасностью потоков |
| Скорость | Медленнее, чем Rust | Лучше всего подходит для высокоскоростных приложений |
| Поддержка | Обширное и разнообразное сообщество и исчерпывающая документация | Сообщество меньше по сравнению с Python, но имеет обширную документацию |

Rust против Python: что вы должны использовать?

Выбор Rust против Python полностью основан на том, что вы будете разрабатывать с использованием этих двух языков программирования. Оба языка имеют свои плюсы и минусы. Как вы можете видеть из этой статьи, Rust является наиболее

подходящим, если вы хотите создавать высокопроизводительные, ресурсоемкие и масштабируемые приложения. Например, это идеальный язык программирования для встраиваемых систем и разработки программ командной строки. Кроме того, когда дело доходит до [интерфейсной разработки](#), существуют лучшие языки, такие как React и Vue, которые более популярны, чем Python или Rust. Однако, хотя Python не так безопасен и эффективен, как Rust, из-за его более быстрой разработки и широкого набора доступных библиотек, вы должны выбрать его для создания прототипов, написания сценариев, обработки данных, машинного обучения и анализа данных. задачи.

Многие компании используют его для анализа данных, а не для веб-разработки. Кроме того, если вы начинающий программист, который не имеет представления ни о каком языке или хотите быстрее [стать веб-разработчиком](#), тогда вы должны выбрать Python, а не Rust, поскольку его легко изучать и программировать на нем.

Краткие сведения

Python – самый популярный язык программирования для самых разных целей. Хотя Rust моложе Python, он становится все более популярным благодаря своим преимуществам перед Python. Ведущие технологические компании мира в значительной степени используют как Python, так и Rust. Безопасность потоков и памяти, а также абстракции с нулевыми затратами являются отличительными особенностями Rust.

В отличие от этого, Python популярен благодаря своему простому синтаксису, динамической типизации, широкой поддержке сообщества и множеству библиотек. Медленное время компиляции и недостаточная поддержка исправления monkey являются основными недостатками Rust. Основными недостатками Python являются низкая скорость и низкая поддержка подключения к базе данных и мобильной разработки. В [то время как Rust и Python можно использовать для программирования общего назначения, только один из них идеально подходит для вашего проекта...](#)

Python – явный победитель в отношении простоты использования, возможностей трудоустройства и [заработной платы разработчиков приложений](#), расширяемости, популярности и поддержки. Однако, хотя Rust – это новый язык, он лучше, чем Python, если учесть обработку ошибок, сбор мусора, управление памятью,

производительность, масштабируемость, безопасность и скорость. В целом, Rust – гораздо лучший язык, чем Python, и имеет большее конкурентное преимущество перед Python, чтобы стать одним из доминирующих языков программирования в мире. Следовательно, это один из [лучших языков программирования, который вам следует изучить](#).

Какой язык — Rust против Python — вы выберете для своего следующего проекта? Дайте нам знать в разделе комментариев.

Дата Создания

01.02.2023