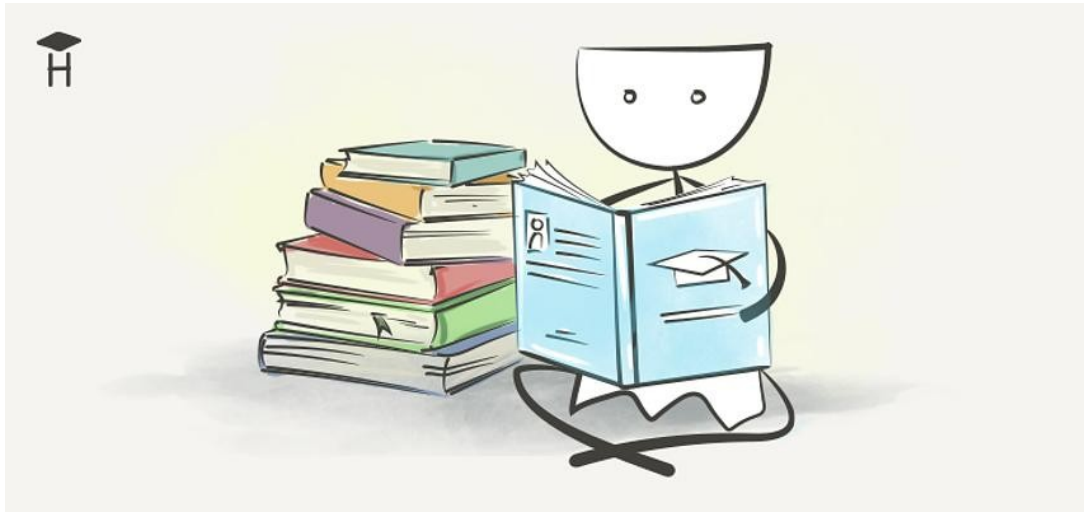


# Python шагает по планете: как сайд-проект стал одним из самых популярных языков программирования в мире



Программист Гвидо ван Россум придумал Python, так как ему не нравились существующие языки программирования. В интервью для ZDNet ван Россум рассказал историю создания «пайтона», а также о его перспективах.

## Пролог

В конце 1994 года группа программистов из разных уголков США собралась, чтобы поговорить о своём новом секретном оружии. В числе участников первого воркшопа по недавно представленному языку Python был инженер Барри Варшава (Barry Warsaw). На встрече присутствовали около 20 разработчиков, и все они были в восторге от нового инструмента.

Барри вспоминает слова одного из участников воркшопа: «Пожалуйста, никому не говорите, что я был здесь, пусть Python будет моим секретным оружием в конкурентной борьбе». По словам Варшавы, уже в 1994 году было понятно, что «пайтон» открывает новые возможности перед программистами, и что писать код и решать практические задачи с помощью этого языка проще и удобнее.

«Это какое-то сочетание читабельности и удовольствия от работы», — так Барри Варшава делится первыми впечатлениями от Python.

Участники первого воркшопа не сохранили секрет. Сегодня число энтузиастов Python намного превышает 20 человек, которые радовались появлению «пайтон» в 1994. Этот язык входит в число самых популярных, у него хорошие перспективы, а специалисты по Python **востребованы на рынке труда**.

Python — язык общего назначения. Он применяется в разных сферах, включая веб-разработку, Machine Learning, Data Science. Этот язык используют крупные компании, правительственные организации, образовательные учреждения. Например, Python применяют в NASA, Netflix и Instagram.

## Python: ранние годы

Python стал популярным и востребованным так же быстро и неожиданно, как британская комик-группа Monty Python, в честь которой был назван язык.

«Пайтон» создал программист Гвидо ван Россум из Нидерландов. В конце 80-х годов прошлого века Гвидо работал в Центре математики и информатики в Амстердаме. Ван Россуму не нравились существующие языки программирования, поэтому он стал работать над новым. Гвидо хотел сделать одновременно простой и мощный инструмент.



*Создатель «пайтона» Гвидо ван Россум*

---

Для постороннего наблюдателя идея создания своего языка программирования может показаться похожей на идею самостоятельно собрать самолёт. Однако Гвидо ван Россум, которому в конце 80-х было около 30 лет, начинал не с нуля. В течение трёх лет в Центре математики и информатики он работал над ABC — интерпретируемым языком программирования. Поэтому Гвидо представлял, как разрабатывать интерпретатор команд и синтаксис.

Важно понимать сложности, с которыми сталкивался ван Россум в конце XX века. Он работал в операционной системе Amoeba, которая требует использования оболочки C или Unix. У этих оболочек есть серьёзные ограничения. Например, C требует ручного управления памятью. Также в ней не хватает библиотек, поэтому программисту нужно заново изобретать велосипед в каждом проекте. А главное ограничение командной оболочки Unix — низкая скорость работы, из-за которой специалисту трудно реализовать сложную логику.

Ограничения этих языков были настолько серьёзными, что создание собственного языка не казалось ван Россуму безумной идеей. «Я подумал, почему бы не взять лучшие идеи из ABC и не создать более простой и удобный язык программирования», — вспоминает Гвидо ван Россум. Он позаимствовал название у комик-группы Летающий цирк Monty Python и стал работать над языком в конце 1989 года.

«В те дни я редко выходил из дома, так как в основном смотрел телевизор и писал код», — рассказывает Гвидо. Кстати, ван Россум оказался в хорошей компании. В конце 80-х мир увидел несколько языков программирования, которые появились благодаря недовольству инженеров инструментами, которые были им доступны. Так Ларри Уолл создал Perl, а Джон Оустерхаут разработал Tcl.

В течение трёх месяцев ван Россум создал работающий прототип Python. Гвидо говорит, что Python сразу был похож на сегодняшний вариант языка, хоть в нём не хватало важных абстракций, которые реализованы сегодня.

«Все базовые компоненты интерпретатора и языка работали, первые простые программы на Python наверняка работают и сегодня», — утверждает ван Россум. По словам инженера, функции в конце 80-х определялись так же, словари и кортежи имели такой же синтаксис, отступы были такими же.

Коллеги ван Россума сразу же стали использовать первую версию Python. Но Гвидо сомневался, что язык станет популярным. В доинтернетовскую эпоху, когда онлайн-курсов не существовало, было сложнее популяризировать и изучать языки программирования.

Сегодня каждый желающий может опубликовать обучающий курс на специализированной платформе, например, на Code Basics, в несколько кликов. А в конце 80-х и начале 90-х популяризировать языки программирования было сложнее. Гвидо ван Россум столкнулся с этим, ещё когда работал над ABC.

«Когда я впервые приехал в США примерно в 1985 году, в моём багаже были кассеты с магнитной лентой», — рассказывает Гвидо. В блокноте у ван Россума были адреса и телефоны людей, которые заинтересовались ABC. Гвидо встречался с ними и демонстрировал возможности языка с помощью кассет.

Развитие интернета помогло ван Россуму популяризировать Python без чемодана с кассетами.

Гвидо ван Россум представил Python в новостных группах alt.sources в 1991 году. Это фактически был opensource-проект, хотя термин open source впервые использовали только через 6 лет. Пользователям Usenet приходилось загружать файл с интерпретатором Python целую ночь. Но это уже было намного эффективнее походов от двери к двери с кассетами.

Ван Россум постоянно получал обратную связь от специалистов, которые использовали Python. Он учитывал их предложения во время работы над новыми версиями языка.

## Почему Python победил

К середине 90-х Python стал популярным языком программирования. Это воодушевляло Гвидо ван Россума. Специалист считал, что другие программисты используют «пайтон» по той же причине, которая привела Гвидо к идее создать новый язык программирования. Они хотели работать с высокоуровневым языком, который не имеет ограничений командной оболочки Unix, но в то же время остаётся простым.

Барри Варшава утверждает, что Python уже в 90-е был сбалансированным языком: с одной стороны простым и удобным, а с другой мощным. По мнению Варшавы, в те времена подобных качеств не было ни у одного языка программирования, кроме «пайтона».

«Я работал с Perl, C и Tcl, и это не доставляло мне удовольствия», — говорит Барри. Он продолжает: «Когда появился Python, я понял, что значит наслаждаться написанием кода».



*Барри Варшава на первом воркшопе по Python в 90-е*

---

Уже в первые годы существования «пайтон» выделялся чистым и понятным синтаксисом. Группировка блоков кода с помощью отступов привлекала программистов в 90-е годы и привлекает их по сегодняшний день.

Представитель исследовательской компании Gartner Финтан Райан (Fintan Ryan) отмечает, что чистота синтаксиса сыграла едва ли не основную роль в популяризации Python.

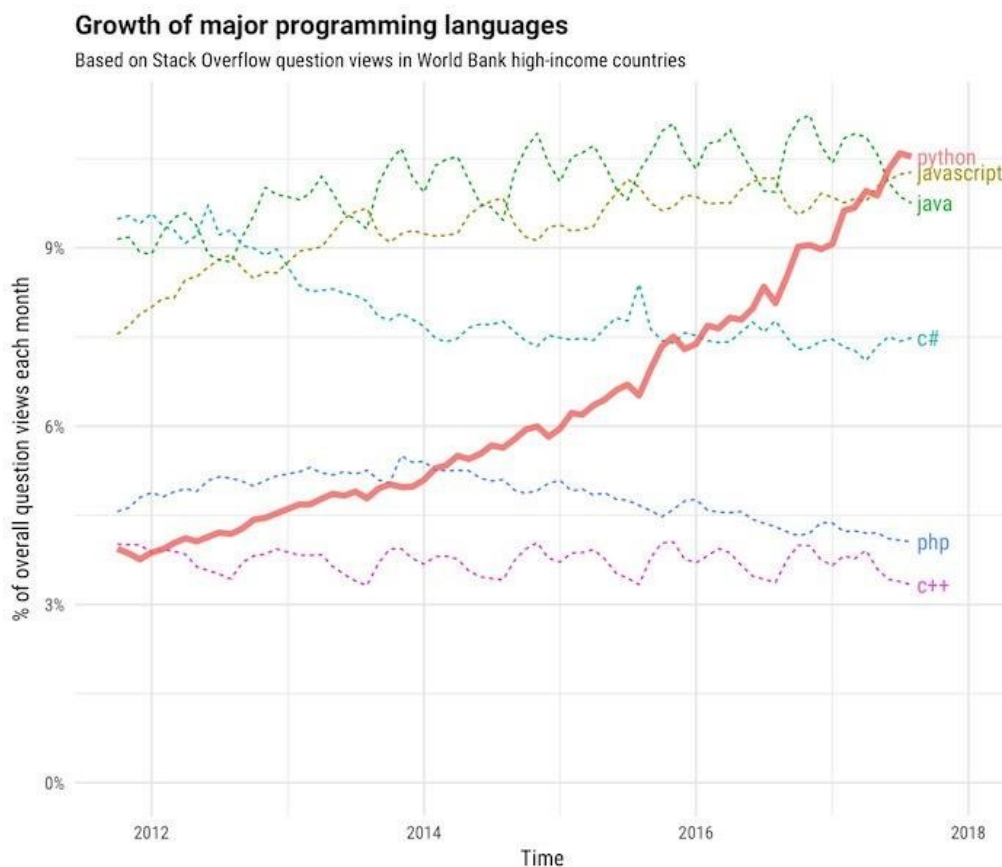
По словам Гвидо ван Россума, чистый синтаксис Python появился не случайно. Гвидо подчёркивает, что программирование — это не только создание инструкций для машины. Это ещё и обмен идеями между разработчиками. Поэтому код должен быть понятным людям.

В свою очередь, Финтан Райан подчёркивает, что Python с начала 90-х предлагал программистам не только чистый синтаксис. «В этом языке были реализованы классы, обработка исключений, lambda-функции, функции высших порядков map и filter», — говорит Финтан.

По словам ван Россума, Python мог не появиться, если бы существующие в 80-е годы языки были более совершенными. Гвидо нравился Perl, но он не мог использовать его в операционной системе Amoeba. «Если бы не эта несовместимость, я бы никогда не задумался о создании нового языка программирования», — говорит ван Россум.

Python в середине 90-х завоёвывал популярность и признание, но всё равно отставал от самых востребованных языков. Главными конкурентами «пайтона» были Perl и Tcl/Tk. Разработчики этих языков тоже стремились к простоте использования и широким возможностям.

По данным [StackOverflow Developer Survey](#), в настоящее время Python развивается быстрее других языков программирования. Популярность Perl упала настолько сильно, что этот язык попадает не во все исследования. На иллюстрации ниже визуализирован феноменальный рост популярности «пайтона».



*Рост популярности Python*

---

Как объяснить разную судьбу Perl и Python? Гвидо ван Россум считает секретом успеха «пайтон» простоту работы с кодом относительно больших приложений. «Perl был удобным для программ, в которых 10 строк кода», — вспоминает Гвидо. По его словам, когда дело касалось приложений, в которых 500 строк кода и тысячи строк библиотек, программисту требовались дополнительные усилия, чтобы поддерживать такую программу на Perl. «С Python такой проблемы нет, код всё время остаётся читабельным, и программисту не приходится тратить слишком много времени на его поддержку», — утверждает ван Россум.

По словам ван Россума, в начале и середине 90-х разработчики нуждались в инструменте, который позволял писать большие приложения без дополнительных усилий. Здесь Python превосходил Perl, C, C++ и Java.

Гвидо ван Россум в середине 90-х ещё работал в Центре математики и информатики. Но Python уже набрал обороты, поэтому Гвидо много времени тратил на общение с разработчиками из разных стран мира.

## Python и интернет

К середине 90-х программисты использовали «пайтон» для решения разных задач, включая программы для создания аудиозаписей и первые попытки веб-разработки, которая в дальнейшем стала одной из главных сфер применения языка. Гвидо ван Россум отмечает, что его сильно интересовал опыт применения Python в веб-разработке.

А Финтан Райан вспоминает, что с помощью Python программисты могли быстро создавать сложные программы. По словам эксперта, этот язык снизил порог входа в разработку в целом и веб-разработку в частности. Райан подчёркивает универсальность Python: «Системные администраторы использовали новый язык для автоматизации систем, разработчики полюбили его за возможность писать код в функциональном стиле». По мнению Финтана Райана, Perl в этом смысле сильно уступал «пайтону».

В 1994 году на Python обратил внимание Майкл Маклэй из Национального бюро стандартов США (NBS). Сегодня это ведомство называется Национальным институтом стандартов и технологий. Маклэй увидел перспективы использования «пайтона» в NBS. По словам ван Россума, в то время в этой организации было «очень много данных, но не хватало программистов, чтобы их обрабатывать».

Майкл Маклэй пригласил Гвидо ван Россума на работу в NBS в качестве приглашённого эксперта от Центра математики и информатики. По задумке специалистов, этот шаг должен был помочь продать Python руководству NBS. Решение Маклэя сыграло важную роль в жизни ван Россума и судьбе «пайтона». К слову, первый воркшоп по этому языку состоялся именно в офисе NBS.



*Гвидо ван Россум, Барри Варшава и Роджер Массе*

В NBS ван Россум познакомился с Бобом Каном, который участвовал в разработке протокола TCP/IP. Кан пригласил ван Россума на работу в Корпорацию национальных исследовательских инициатив (CNRI). Это неправительственная организация, которая занималась исследованием и развитием информационных сетей. Это приглашение пришлось кстати, так как ван Россум в то время серьезно задумывался о своём будущем в Центре математики и информатики.

По словам Гвидо, Центр математики и информатики был научным учреждением, поэтому руководство ставило специалиста перед необходимостью получения степени PhD. Ван Россум предпочитал развивать Python, а не заниматься научными исследованиями.

В апреле 1995 года Гвидо ван Россум присоединился к команде CNRI. Здесь он разрабатывал программы Knowbot. Команда работала на Python. В CNRI ван Россум сотрудничал с Джереми Хилтоном, Роджером Массе, Барри Варшавой, Кеном Манхаймером, Фредом Дрейком. Эти люди стали ядром сообщества Python и сыграли важную роль в развитии языка.

Вместе с новыми друзьями Гвидо ван Россум открыл сайт [python.org](http://python.org). На этом ресурсе хранилось ядро Python. Также сайт помогал координировать работу специалистов, которые участвовали в развитии «пайтона».

Уже во второй половине 90-х вокруг Python сформировалось многочисленное сообщество. Чтобы формализовать управление языком, Гвидо ван Россум и его

друзья создали прообраз Python Software Foundation. Окончательно эта организация оформилась в 2001 году. Регулярные воркшопы по Python превратились в ежегодные конференции PyCon.

К началу XXI века Python развивался преимущественно благодаря вкладу сообщества. Программисты уже не шутили на тему «что будет с языком, если ван Россум отойдёт от дел по непредвиденным обстоятельствам». Тем не менее, Гвидо играл ключевую роль в развитии языка. К нему даже приклеился ярлык «доброжелательный диктатор». Он отлично иллюстрирует стремление ван Россума участвовать в ключевых вопросах развития «пайтона».

Финтан Райан из Gartner подчёркивает, что стремление создателя языка контролировать его развитие — нормальное явление. Точно так же вели себя создатели Perl Ларри Уолл и создатель Node.js Райан Дал.

Сам Гвидо ван Россум считает, что важным фактором успеха «пайтона» стала именно открытость. По его словам, ключевые вопросы развития языка всегда обсуждались в сообществе.

## Эволюция Python

В следующие годы Python быстро развивался. Тектонические изменения произошли в 2008 году, когда был представлен Python 3.0.

В 2018 году Гвидо ван Россум оставил неформальный пост доброжелательного диктатора, то есть перестал руководить разработкой языка. Это решение связано с нежеланием сообщества имплементировать предложение ван Россума об использовании в «пайтоне» выражений присваивания. Гвидо считал, что новый инструмент ускорит Python. Представители сообщества посчитали, что выражения присваивания снизят читабельность кода на Python.

По словам ван Россума, он привык к обсуждениям нововведений. А в отставку он ушёл из-за агрессивной манеры ведения спора некоторых оппонентов, а также из-за перехода критиков на личности.

«Это не было революцией, но я почувствовал, что ключевые люди в сообществе мне не доверяют», — прокомментировал ситуацию Гвидо ван Россум. По мнению Гвидо, дискуссия, которая привела к его отставке, стала возможной благодаря популярности Python.

Позже сообщество избрало Гвидо ван Россума, Барри Варшаву, Брэтта Кэннона, Кэрл Виллинг и Ника Коглана в наблюдательный совет, который следит за развитием языка. По словам Барри Варшавы, эти перестановки помогут управлять развитием Python в условиях стремительного роста популярности этого языка программирования.

Барри Варшава отмечает, что 25 лет назад Гвидо ван Россум самостоятельно справлялся с ответственностью за будущее «пайтона». «В текущих условиях это слишком большая нагрузка на одного человека», — говорит Варшава.

По словам Варшавы, выборы в наблюдательный совет будут проводиться после каждого серьёзного обновления «пайтона». «Если мы хотим, чтобы Python



развивался так же успешно в следующие 25 лет, управлять процессами должны молодые лидеры, а не я и Гвидо», — подчёркивает Барри Варшава.

Мариатта Виджая, один из ключевых членов сообщества, считает, что ситуация развивается правильно. «Наблюдательный совет лучше справится с развитием языка, чем один человек», — подчеркнула Мариатта.



*Мариатта Виджая*

---

## Будущее Python

Популярность «пайтона» растёт, по состоянию на начала осени 2019 года он входит в число самых востребованных языков программирования. Но опытные представители сообщества уже говорят о вызовах, с которыми столкнётся язык в ближайшее время. Python должен преодолеть эти вызовы, чтобы оставаться популярным.

Например, на одной из конференций сооснователь BeeWare Рассел Кит-Магги во время выступления сказал, что у «пайтона» начнутся серьёзные проблемы, если в ближайшее время не улучшится поддержка мобильных и современных веб-платформ.

«Смартфонами и планшетами пользуется практически каждый человек, ноутбукам и настольным компьютерам такое не снилось», — говорит Кит-Магги. По его словам, сообществу разработчиков Python надо задуматься об использовании языка на девайсах этого типа.

Рассел Кит-Магги указал и на другие проблемы, требующие внимания сообщества. Среди основных:

- инструменты тестирования Python некорректно работают на мобильных девайсах;
- установочные файлы приложений на «пайтоне» слишком «тяжёлые»;
- для корректной работы библиотеки `asyncio` на разных платформах требуются дополнительные усилия;
- многие модули стандартных библиотек невозможно использовать с интерпретаторами, отличными от CPython.

Барри Варшава поддерживает опасения Кит-Магги. По мнению Варшавы, сообщество Python должно уделить внимание совместимости языка с планшетами и смартфонами, а также с современными технологиями, например, WebAssembly.

Варшава считает важным «научить» Python эффективно распределять задачи между ядрами процессоров. Барри подчёркивает важную роль подинтерпретаторов Python, над которыми работает Эрик Сноу. Эти подинтерпретаторы обеспечивают возможность исполнять код на «пайтоне» на разных ядрах процессора.

Разработчики Python модернизируют стандартную библиотеку. Недавно сообщество предложило удалить из неё устаревшие модули. Стандартная библиотека считается одной из сильных сторон «пайтона». Благодаря ей о Python говорят, что у него «батарейки в комплекте». Однако на Python Language Summit 2019 активно обсуждалась возможность выбора библиотек из репозитория PyPI. Теоретически такая возможность обеспечила бы программистам большую свободу выбора инструментов.

В сообществе Python активно обсуждается возможность расширения органов управления. Этот шаг позволил бы отразить разнообразие участников сообщества, в частности, Мариатта Виджая отмечает, что хотела бы привлечь в сообщество PyLadies больше представителей из Индии и Африки.

Барри Варшава говорит, что развитием Python в большей степени управляет сообщество, несмотря на разговоры о доброжелательной диктатуре. Примером созданного сообществом проекта, который сильно повлиял на «пайтон», стал Type Hints. Этот инструмент позволяет проверять типы и уменьшать количество ошибок во время работы. Ещё один пример созданного сообществом инструмента — упомянутая выше библиотека `asyncio`.

Гвидо ван Россум тоже уверен, что сообщество будет главным двигателем дальнейшего развития Python. «У нас очень сильное сообщество и новая система управления, поэтому мы готовы к вызовам, с которыми столкнётся язык», — говорит ван Россум.