



semen\_grinshtein 17 сентября 2016 в 18:25

# 15 самых популярных языков программирования по версии GitHub

JavaScript, Java, C++, C, C#



Разработчики программного обеспечения пользуются огромным спросом в настоящее время. В некоторых компаниях даже стажеры-программисты получают высокую зарплату. ИТ-компании борются друг с другом за талантливые кадры.

А кадры, в свою очередь борются за место под солнцем. Успех тех и других будет зависеть от того, насколько хорошо у них получается держать руку на пульсе, быть в тренде, использовать перспективные технологии и языки программирования. Чтобы понять, на что ориентироваться и в каком направлении идти, разработчики ПО и их работодатели изучают различные исследования и рейтинги популярности – будь то бизнес-модели или те же технологии и языки программирования.

Однако некоторые игроки ИТ-рынка оказались предприимчивыми и создали проекты, посвященные ранжированию и трендам. К примеру, на этой неделе GitHub опубликовал собственный рейтинг 15 самых популярных языков программирования. Конечно же, популярность определялась по количеству pull-запросов на GitHub за последний год.

В настоящее время в GitHub представлены 5.8 миллиона активных пользователей, 331 тысяча организаций и 19.4 миллиона репозиторий.

## No. 15 — TypeScript:

Язык своим появлением обязан компании Microsoft. Он создан как средство разработки веб-приложений, расширяющее возможности JavaScript. Основным разработчиком языка TypeScript является Андерс Хейлсберг который ранее участвовал в создании Turbo Pascal, Delphi и C#. В ближайшее время выйдет релиз TypeScript 2.0.

## No. 14 — Swift:

Язык собственного приготовления – от компании Apple. Он был создан для iPhone-приложений в 2014 году. Однако Swift за столь короткий срок сумел завоевать популярность.

Такси-сервис Lyft переписал свое iPhone-приложение на этом языке и сообщил об «огромном скачке» в производительности.

А совсем недавно вышло новое приложение Swift Playgrounds – самый простой способ познакомиться с языком Swift. По

крайней мере, в этом уверен Крейг Федериги, старший вице-президент компании Apple по программному обеспечению.

### No. 13 — Scala:

Этот язык программирования вышел в свет в 2001 году. Его подхватили такие крупные компании, как Airbnb и Apple. По их мнению, на нем проще и быстрее писать приложения для их нужд, нежели на набившем оскомину языке Java.

Изначально Scala, поддерживающий объектно-ориентированную и функциональную парадигмы, был создан с расчетом на трансляцию в байт-код Java и .NET. Со временем также появился транслятор Scala в код JavaScript — Scala.js. Однако в мае 2016 года стало известно, что в рамках проекта Scala Native создается компилятор, обещающий ускорить выполнение приложений, написанных на этом языке.

### No. 12 — Objective-C:

Изначальный C так понравился людям, что у него появились последователи. В частности, некоторые из них вдохновились на создание Objective-C, который предстал перед публикой в 1983 году. Конечно, в него были добавлены новые идеи и элементы языка Smalltalk, но буква C не зря венчает его название.

Особенно распространился Objective-C среди разработчиков под платформы Apple. Пока он удерживает эти позиции, но Swift грозит потеснить потомка C в скором времени.

### No. 11 — Shell:

Это гадкий утенок среди прочих участников списка: Shell – не столько язык, сколько интерпретатор команд для выполнения тех или действий в операционных системах семейства UNIX. Его скрипты используются, например, для автоматизации обновления ПО. Он содержит стандартные конструкции для циклов, ветвления, объявления функций и так далее.

### No. 10 — Go:

Был разработан внутри компании Google. Первоначальная разработка Go началась в сентябре 2007 года, а его непосредственным проектированием занимались Роберт Гризмер, Роб Пайк и Кен Томпсон. Официально язык был представлен в ноябре 2009 года.

Язык Go разрабатывался как язык системного программирования для создания высокоэффективных программ, работающих на современных распределённых системах и многоядерных процессорах. Он может рассматриваться как попытка создать замену языку Си.

При разработке уделялось особое внимание обеспечению высокоэффективной компиляции. Программы на Go компилируются в объектный код (хотя доступен и интерпретатор) и не требуют для исполнения виртуальной машины.

### No. 9 — C:

Это язык, которому теперь пытаются найти замену. Язык Си был разработан Деннисом Ритчи в 1972 году в Bell Labs. Он является предшественником таких языков программирования как C++, Java, C#, JavaScript и Perl. По этой причине изучение этого языка ведет к пониманию и других языков. Язык C используется для разработки низкоуровневых приложений, так как считается ближе всего к аппаратному, уступая только ассемблеру.

### No. 8 — C#:

C# принадлежит семье языков программирования Microsoft и был разработан в 2000 году и стал частью первого релиза .NET framework. Язык C# сочетает в себе надежность C++ с дополнительными возможностями Java. Поэтому если вы хорошо знаете Java, можно легко переключиться на C# и наоборот.

Язык C# позволяет разрабатывать практически любые приложения, которые связаны с Visual Studio IDE.

### No. 7 — CSS:

Cascading Style Sheets (каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам.



Указан рост популярности языка в процентах. Его место в рейтинге определяется числом pull-запросов пользователей. Рейтинг отражает популярность применения языков программирования в репозиториях.

#### No. 6 — C++:

Языку Си не хватало работы с объектами. Чтобы решить проблему, в 1986 году был разработан C++, который является одним из самых распространенных языков в мире. Google Chrome, Mozilla Firefox, Winamp и линейка продуктов Adobe были разработаны с помощью C++. Кроме того, некоторые современные игры и операционные системы были разработаны на C++ из-за быстрого процессинга и компиляции. Кроме того, разработчики C++ имеют огромный спрос на рынке труда.

#### No. 5 — PHP:

PHP — это один из наиболее широко используемых языков для разработки динамических веб сайтов. PHP был разработан в 1995 году и является языком сценариев на стороне сервера, что означает, что PHP код обрабатывается на сервера, а конечный результат пользователь получает в виде обычного HTML.

PHP открытый язык разработки, поэтому написаны уже тысячи модулей, которые можно модифицировать до требуемой функциональности.

Однако недоброжелатели не дремлют: Джефф Этвуд, основатель Stack Exchange, как-то писал, что PHP – это даже не язык программирования, это куча не связанных друг с другом фрагментов функциональности.

#### No. 4 — Ruby:

Ruby — простой и читаемый язык программирования, ориентированный на разработку веб приложений. Разработанный Юкиhiro Мацумто в 1995 году, фреймворк Ruby использовался для разработки Github, Scribd, Yammer, Shopify и Groupon. Ruby сочетает в себе некоторые возможности Lisp, Pearl и Eiffel. Ruby имеет хороший рынок труда и разработчики получают достойно.

#### No. 3 — Python:

Python — это другой высокоуровневый язык программирования и часто считается самым легким языком, благодаря своей простоте, читаемости и синтаксису. Python был разработан Гвидо Ван Россумом в 1991 году. Python не использовался так широко в прошлом, однако стал особенно популярным за последние годы благодаря инвестициям Google. В настоящее время некоторые весьма известные и надежные сайты работают на python, особенно pinterest.com, instagram.com и rdio.com. Как и

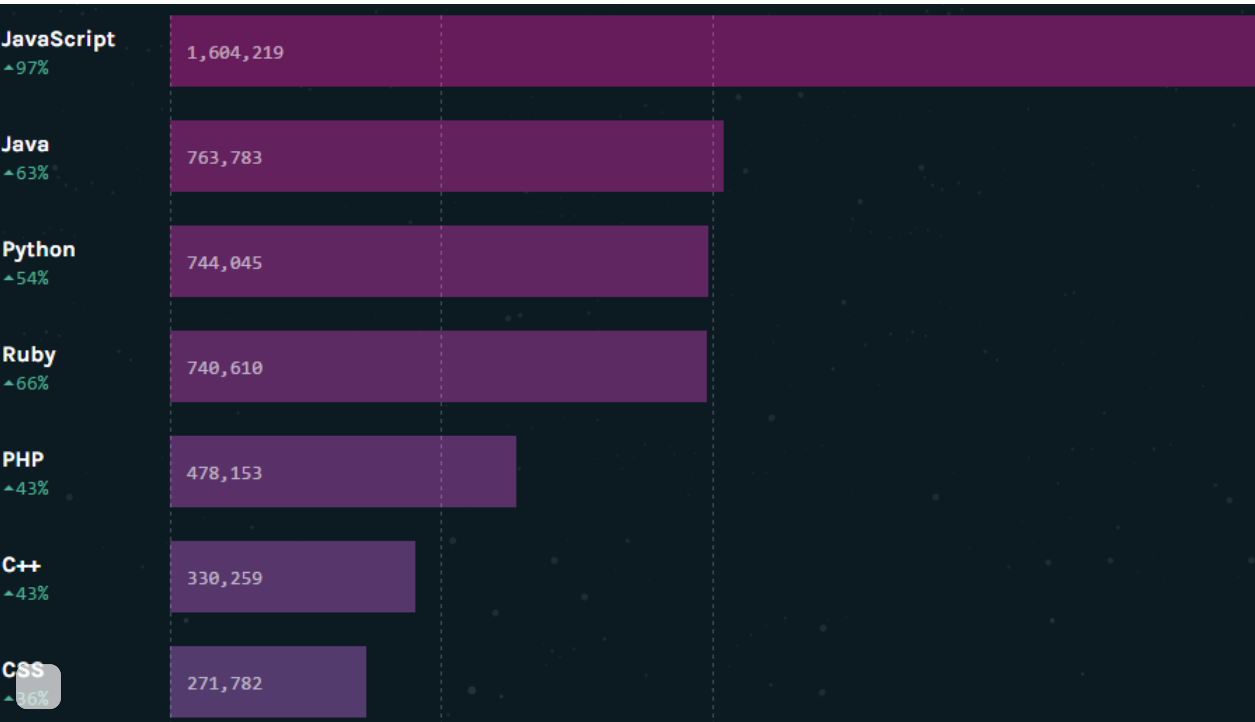
PHP, Python можно использовать для разработки веб приложений.

No. 2 — Java:

Java был разработан Джеймсом Гослингом в 1990 году в компании Sun Microsystems. Java дополнительно увеличивает возможности языка C++. Особенность Java заключается в том, что это первый чисто объектно-ориентированный язык программирования. Java был разработан по принципу WORA (Write Once Run Anywhere или «написав код однажды, вы запустите его везде»). Речь идет про переносимость Java. Необходимо скомпилировать исходный код на Java всего лишь раз, а затем запустить на любой машине с установленным JVM (Java Virtual Machine) и затем использовать.

No. 1 — JavaScript:

Серверные языки сценариев идеально подходят для разработки сложных веб приложений, но каждая такая задача сильно нагружает сервер. Поэтому разработчики делегировали часть функций на сторону клиента и использовали JavaScript. JavaScript — это язык программирования, выполняемый в клиентском браузере и обрабатывает команды на компьютере конечного пользователя, а не сервера, что приводит к снижению нагрузки на сервер и увеличению скорости работы приложения. JavaScript был разработан компанией Netscape и вряд ли есть сайты, которые не используют его.



Указан рост популярности языка в процентах. Его место в рейтинге определяется числом pull-запросов пользователей. Рейтинг отражает популярность применения языков программирования в репозиториях.

По данным GitHub, JavaScript, C# и Go показали двойной рост аудитории. А аудитория Swift и TypeScript выросла в 3,5 раза.

Индекс TIOBE распределил места немного по-другому: на первом месте оказался языка Java, далее следуют C, C++, C# и Python. А лидер рейтинга GitHub здесь занимает только 6-е место.

Sep 2016	Sep 2015	Change	Programming Language	Ratings	Change
1	1		Java	18.236%	-1.33%
2	2		C	10.955%	-4.67%
3	3		C++	6.657%	-0.13%
4	4		C#	5.493%	+0.58%
5	5		Python	4.302%	+0.64%
6	7	▲	JavaScript	2.929%	+0.59%
7	6	▼	PHP	2.847%	+0.32%

Теги: разработка по, github, рейтинг, языки программирования, си, javascript, python, java, swift

Хабы: JavaScript, Java, C++, C, C#

↑ +31 ↓ 111 186k 112 Поделиться



Гриша Гринштейн @semen\_grinshtein

Пользователь

#### ПОХОЖИЕ ПУБЛИКАЦИИ

4 января 2011 в 19:13

Стековый язык программирования на JavaScript в 34 строки

↑ +23 1,9k 19 9

20 октября 2010 в 20:33

Рейтинг языков программирования (октябрь, 2010 год)

↑ +28 6,7k 14 112

15 января 2010 в 12:49

Январский рейтинг языков программирования

↑ +37 13,9k 9 113

#### ВАКАНСИИ

Ведущий инженер по автоматизации тестирования

от 170 000 до 300 000 Р • RH Partners • Можно удаленно

Java-разработчик

до 150 000 Р • Логика Бизнеса • Можно удаленно

Наставник онлайн-курсов по JavaScript

от 12 500 до 20 000 Р • Hexlet • Можно удаленно

Java-разработчик

от 100 000 Р • Gaskar Group • Москва • Можно удаленно

Middle Java Developer (Заказная разработка)

от 160 000 до 200 000 Р • Innovative People • Москва • Можно удаленно

Больше вакансий на Хабр Карьере

#### Комментарии 112



qrc13 17 сентября 2016 в 18:34

↑ +13 ↓

Почему самый костыльный и ужасный язык занял позицию #1?



xtala 17 сентября 2016 в 18:47

↑ +11 ↓

Да не такой уж он костыльный и ужасный. Готовить его надо уметь и повар хороший нужен. А на первом месте наверное потому, что веб становится де-факто самой большой программной экосферой из всех существующих.



maaGames 17 сентября 2016 в 18:54



↑ +27 ↓

Потому что это не «популярность», а «количество коммитов». А это не одно и то же.

Не говоря о том, что многие разработчики C++ и прочих старых языков не бросаются переводить все проекты на github из VCS, которыми пользовались до этого кучу лет.



Inroma 17 сентября 2016 в 20:23



↑ -11 ↓

Как рас потому что на нём много пишут кода ( в смысле объёма )



ivan386 17 сентября 2016 в 20:29



↑ -8 ↓

Я недавно worker'ы осваивал а их на локальной машине не потестить не поднимая вебсервер. Такого количества комитов в один день я ещё никогда не делал.



samizdam 19 сентября 2016 в 07:41



↑ 0 ↓

Почему же?



ivan386 19 сентября 2016 в 13:55



↑ 0 ↓

Что именно?

```
gravity.svg:28 Uncaught SecurityError: Failed to construct 'Worker': Script at 'file:///C:/.../worker.js' cannot be accessed from origin 'null'.
```

1. Пишем код
2. Commit
3. Push
4. Тест на сервере
5. Исправляем, дописываем
6. GoTo 2



samizdam 19 сентября 2016 в 14:23



↑ +1 ↓

Я к тому, что не проблема поднять локально вебсервер средствами docket / vagrant. Это сэкономит время отладки и избавит от мусорных коммитов в origin.



ivan386 19 сентября 2016 в 15:07



↑ 0 ↓

Это не выглядит как что то простое. Я попробовал поискать простой веб сервер но тот что мне попался вместо открытия файла заставлял браузер скачивать. Я забил и тестил на rawgit.

Согласен это не является хорошей практикой но так случилось.



gearbox 19 сентября 2016 в 20:26



↑ +1 ↓

если стек js — express, там для статики пять строчек. Либо node-static — <https://learn.javascript.ru/screencast/webpack> во второй части (простая сборка, простой конфиг) Можно использовать встроенный php сервер <http://php.net/manual/en/features.commandline.webserver.php> В общем как бы варианты есть.



Nicklasos 20 сентября 2016 в 12:12



↑ +1 ↓

Просто устанавливаем php и в консолю выполняем команду `php -S localhost:8000` для запуска локального веб сервера.

Для https можно установить ngrok и запустить его командой `ngrok http 8000` и у вас локальный https веб сервер без

всяких проблем.

Думаю у `node.js` в пакетах где-то тоже должен быть простой веб сервер.



**Alexeyslav** 21 сентября 2016 в 10:11



↑ 0 ↓

Так вестимо, браузер настраивать надо. Это поведение зависит от того как ответит сервер, значением в поле тип контента — если он на всё будет отвечать «двоичные данные» то у браузера не останется иного варианта как просто скачивать содержимое как неизвестного типа. Нужно добиться настройкой сервера чтобы тот отвечал на изображение соответственно, тогда браузер откроет его как изображение. Расширение файла роли не играет — имени файла так вообще может не быть.



**EmmGold** 17 сентября 2016 в 20:37



↑ +14 ↓

Вероятно часто исправляют исправления на исправления. Поэтому и используется много, и коммитов много.



**gnemod** 17 сентября 2016 в 21:00



↑ -1 ↓

Да он костыльный и ужасный с точки зрения тех кто программирует на «правильных» языках типа Java или C#, но тем не менее почему то программировать на нем мне гораздо приятнее чем на C, C++, C#, Swift (даже не знаю почему, может я люблю говнокодить). А с последними расширениями ECMA он убьет и typescript и займет весомую часть в разработках серверного бэкэнда



**qrcck13** 17 сентября 2016 в 21:01



↑ +5 ↓

Я бы сказал что в JS — порог вхождения заметно ниже. Не нужно много и долго изучать примудрости Computer Science, что бы стать более менее «средним» разработчиком на JS



**gnemod** 18 сентября 2016 в 01:25



↑ -12 ↓

А может не нужны все эти премудрости Computer Science? Когда посредственный программист 1C (iOS, React подставь что хочешь) зарабатывает в полтора раза больше embedded программиста, спрашивается зачем нужны все эти знания архитектуры процессора, математики, паттернов проектирования.



**qrcck13** 18 сентября 2016 в 01:44



↑ +5 ↓

> А может не нужны все эти премудрости Computer Science? Когда посредственный программист 1C (iOS, React подставь что хочешь) зарабатывает в полтора раза больше embedded программиста, спрашивается зачем нужны все эти знания архитектуры процессора, математики, паттернов проектирования.

1. Причем здесь embedded?

2. Я в жизни не видел ни одного посредственного (как вы выразились) 1C / React / etc программиста, котрый получал бы больше квалифицированного разработчика. Любые большие проекты требуют квалификации и опыта, и «посредственные разработчики» просто не способны в них участвовать. А на мелкой халтуре, для которой годиться React, много не заработать.

3. Если вы пошли в разработчики ПО только ради денег — долго вы в профессии не удержитесь



**gnemod** 18 сентября 2016 в 01:58



↑ -3 ↓

1. Embedded имхо требует знания наибольших знаний в области Computer Science
2. Че прям React только для мелкой халтуры годится? Клиентские и серверные javascript фреймворки еще только начали свой путь развития, и глядя на то что уже есть я думаю что кроме javascript уже ничего не надо изучать:) А серьезно WEB разработкой я всего лишь пару лет назад занялся
3. Да чет 20 лет уж там, может и правда пора на покой. Я думаю немного тех кто по молодости может оценить сколько принесет денег та или иная профессия и те или иные скилы. Интересно а ради чего пошли в профессию вы?



**qrcck13** 18 сентября 2016 в 02:53



↑ +4 ↓

> 1. Embedded имхо требует знания наибольших знаний в области Computer Science

Embedded чаще всего требует хорошего знания конкретного железа. Конкретные требования конечно будут разные в зависимости от ситуации. Embedded занимающейся прошивкой ABS блока автомобиля, — это не то же самое, что

embedded разработчик клепающий прошивку к какой-нибудь авто-поливалке. Но тут, насколько я видел по своему опыту, чаще бывает достаточно просто хорошего знания железа, с которым приходится работать.

Все потоки   Разработка   Администрирование   Дизайн   Менеджмент   Маркетинг   Научпоп



Войти

Регистрация

Сам я с ним не сталкивался, но все что читал — говорит о том, что когда нужно что-то аппаратно или OS специфичное, то React уже не годен, и приходится писать все нативно. А React — да, больше для приложений типа «webview на весь экран»

> Клиентские и серверные javascript фреймворки еще только начали свой путь развития, и глядя на то что уже есть я думаю что кроме javascript уже ничего не надо изучать:)

Существует очень много областей, где javascript/веб приложения — не применимы. Конечно если ваша цель — упростить себе жизнь, то конечно, стоит выбрать что проще, и только этим и заниматься :)

> Интересно а ради чего пошли в профессию вы?

Я пошел не в профессию. Я просто занялся тем, что нравилось, оказалось за это еще деньги платят и не малые.



**Nagg** 17 сентября 2016 в 21:33



↑ +8 ↓

займет весомую часть в разработках серверного бэкэнда

Всегда вспоминаю:

[https://twitter.com/phillip\\_webb/status/758824415547109376](https://twitter.com/phillip_webb/status/758824415547109376)



**redmanmale** 19 сентября 2016 в 17:00



↑ 0 ↓

Или лучше: <https://twitter.com/lisovskyvlad/status/393282519438995456>



**lookid** 17 сентября 2016 в 21:50



↑ +11 ↓

Потомучто ядро линукс 1 штука, джава 1 штука, какой-нибудь хибернейт 1 штука. И 100 плагинов для джейквери.



**Odrin** 19 сентября 2016 в 18:35



↑ +1 ↓

Ну Вы еще скажите, что не тащите тонну Java библиотек в любом, более-менее большом, проекте.



**Throwable** 19 сентября 2016 в 19:06



↑ 0 ↓

Если Вы заметите, в IT подобное часто бывает. Применительно практически к любым технологиям: всегда найдется более изящный, но менее аналог.



**xotonic** 20 сентября 2016 в 16:21



↑ 0 ↓

Github Pages

НЛО прилетело и опубликовало эту надпись здесь



**semenyakins** 17 сентября 2016 в 22:29



↑ +6 ↓

Ну как же... А похоловарить на выходных? По-моему стоит вообще подобный блог на хабре сделать — «Холивар». Тогда стало бы ясно куда определять всякие посты вроде «Java vs C++ — кто самый мощный?» или «Самые популярные языки по версии оскоровского комитета». Желаящие могли бы подписываться и отписываться. Все довольны.



**tema\_sun** 17 сентября 2016 в 22:50




↑ +19 ↓

Ну, например, теперь стало ясно, что цсс круче си-шарпа.




↑ ↓



 **xtala** 17 сентября 2016 в 18:40 # 📖


↑ 0 ↓

Еще не заходя в публикацию я знал, что на первом месте будет Javascript. Может оставить заявку на «Битву экстрасенсов»? Типа: «А, сейчас наши участники попытаются определить на каком языке написана эта всепожирающая память утилита для разгадывания гороскопов, что таинственным образом появилась на сайте vasyan.ucoz.ru! » :)

 **Nagg** 17 сентября 2016 в 19:06 # 📖

↑ +1 ↓

Как-то уж сильно статистика гитхаба с Tiobe расходится по Си.

 **svboobnov** 18 сентября 2016 в 16:53 # 📖 🔗 🔄


↑ 0 ↓

Гитхаб же считает пулл-запросы, а Тиобе — количество вопросов в поисковиках + количество вакансий ну, и что-то ещё.

 **lohmatiy** 20 сентября 2016 в 16:22 # 📖 🔗 🔄

↑ 0 ↓


Гитхаб, по всей видимости, считает коммиты. Не умаляя заслуг гитхаба просто скажу, что он отражает только активность разработки в хостящемся на нём Open Source, хотя реальная жизнь гораздо шире, и в этом плане я больше доверяю Tiobe.

 **nxrighthere** 17 сентября 2016 в 19:21 # 📖

↑ +2 ↓


Разве CSS уместен в этом рейтинге?

НЛО прилетело и опубликовало эту надпись здесь

 **SerafimArts** 19 сентября 2016 в 05:37 # 📖 🔗 🔄

↑ 0 ↓

А если взять препроцессоры (диалекты), вроде Sass, то там и до уровня shell не далеко =) Циклы, условия, переменные — всё есть.

 **nolane** 20 сентября 2016 в 16:22 # 📖 🔗 🔄

↑ 0 ↓

Вас слова "формальный язык" ввели в заблуждение или что?

 **zezic** 20 сентября 2016 в 16:20 # 📖 🔗 🔄

↑ 0 ↓

На нём можно написать крестики-нолики, но это займёт несколько тысяч строк кода, *поэтому он смог просочиться в этот список.*


 **Terras** 17 сентября 2016 в 19:24 # 📖

↑ +18 ↓

Все очень просто:

Бекенд для веба можно писать на: c#, java, python, ruby, php, js  
Фронт пишется исключительно на js

Отсюда и получаем, что JS топ-1. Ибо это монополист во фронте сейчас.

 **Infthi** 17 сентября 2016 в 22:38 # 📖 🔗 🔄

↑ 0 ↓

Не исключительно. У нас вот GWT например.

НЛО прилетело и опубликовало эту надпись здесь

 **kryvichh** 20 сентября 2016 в 16:23 # 📖 🔗 🔄

↑ 0 ↓

Мир стал бы очень скучен, если весь фронтенд сводился бы только к Вебу. Да и есть другие языки, кроме перечисленных, на которых можно писать программы и с фронтендом, и с бэкендом, для Windows и Mac. Вот пример — FL Studio написанный полностью от «задницы» до «передницы» в Delphi: <https://www.youtube.com/watch?v=c43Y-FMFDDQ>

 **gearbox** 17 сентября 2016 в 19:33 # 1 ↑ +2 ↓

Perl, ехарный бабай, а не Pearl (Harbor). И странно что сам перл не попал в рейтинг (или он сидит в shell )?

 **PerlPower** 18 сентября 2016 в 04:28 # 1 1 1 ↑ +7 ↓

Он сидит в наших сердцах, причем весьма крепко.

 **jakshi** 20 сентября 2016 в 16:22 # 1 1 1 ↑ +1 ↓

Верное замечание.

Справедливости ради стоит сказать что язык программирования Pearl существует:  
[https://en.wikipedia.org/wiki/PEARL\\_programming\\_language](https://en.wikipedia.org/wiki/PEARL_programming_language)

Но на становление Ruby никоим образом не повлиял.

Когда зная Ruby я начал учить Perl — с удивлением обнаружил множество схожих концепций и даже конструкций в этих двух языках.


 **TimsTims** 17 сентября 2016 в 20:06 # 1 ↑ 0 ↓

А как считали тех, кто пользуется сразу несколькими «языками»?

Те, кто юзает PHP — часто пишут сайты целиком, в т.ч. фронтэндную часть. Значит юзают также CSS, JS и HTML(о боже, где же он тут в рейтинге? неужели html теряет популярность?). Собственно вопрос — как посчитали таких людей?

То же самое можно сказать и про остальные части. Редко, когда разработчики использует только\_один\_язык\_программирования.

НЛО прилетело и опубликовало эту надпись здесь

 **TimsTims** 17 сентября 2016 в 21:39 # 1 1 1 ↑ 0 ↓

Ок. По количеству файлов измененных?

 **Aivendil** 17 сентября 2016 в 22:55 # 1 1 1 ↑ 0 ↓

Во вступлении статьи.


Конечно же, популярность определялась по количеству pull-запросов на GitHub за последний год.

 **Areso** 18 сентября 2016 в 11:56 # 1 1 1 ↑ +1 ↓

Что самое интересное, несколько моих проектов Github'ом распознаются как HTML, хотя там HTML'a 10 строчек и несколько сотен строчек JS.

 **easimonenko** 18 сентября 2016 в 19:50 # 1 1 1 ↑ 0 ↓


Аналогично, а markdown там вообще похоже не учитывается.

 **DarkEld3r** 19 сентября 2016 в 11:13 # 1 1 1 ↑ +1 ↓


Вроде, можно на это повлиять: <https://help.github.com/articles/my-repository-is-marked-as-the-wrong-language/> и дальше по ссылкам.

 **bano-notit** 20 сентября 2016 в 16:20 # 1 1 1 ↑ 0 ↓

И действительно. Всё же мне кажется что можно было считать именно по людям. То есть, в каждом проекте несколько языков. Тот который больше, на том и написан проект. У человека много проектов, если большее количество по предыдущему правилу написаны на плюсах, то считать, что человек пишет больше всего на плюсах. И уже считать количество людей. Мне кажется, что так будет интересней и более точно. НО! Так теряется объективность за год, и мы не узнаем какой язык именно в этом году более интенсивно используется и так мы получается ещё считаем некропользователей.


 **robert\_ayrapetyan** 17 сентября 2016 в 20:23    +8 

>Это язык, которому теперь пытаются найти замену  
Эээ... кто и зачем пытается найти замену C?

 **VioletGiraffe** 18 сентября 2016 в 16:27     -1 

Зачем? Например, затем, что на C крайне сложно писать надёжный и безопасный код. А делать это быстро так и вообще невозможно.



Не хочу начинать холивар, но скажу, что откровенно не понимаю, кто и зачем использует язык C в 2016 году. Не считая legacy для какого-нибудь Z80, под который нет компилятора C++, конечно.

 **semenyakinVS**  18 сентября 2016 в 18:45     +2 

Как плюсовик скажу так... Подозреваю, что минимум для кросскомпиляторного кода. В C нет проблем с декорированием имён и прочими радостями статического и динамического связывания C++-библиотек. Насколько я знаю, по этой причине часто используется бриджинг через C интерфейсов middleware-библиотек.

Помимо этого (в этом уже не так уверен) — под C намного проще делать компилятор и потому компиляторы под C по логике должны выходить раньше C++-компиляторов под всякие зверинцы новых микроконтроллеров. Поэтому C отгрызает часть сегмента Java и вполне себе ничего чувствует себя в этой области.

Люди, пишущие на C, поправьте меня если где-то не прав.

 **VioletGiraffe**  18 сентября 2016 в 19:18     0 

Вы имеете в виду возможность линковать вместе объектные файлы, собранные под одну платформу разными компиляторами?



 **semenyakinVS** 18 сентября 2016 в 20:10     0 

Ага. Эту проблему наследуют статические библиотеки (что весьма логично). С динамическими библиотеками немного другая специфика. Классы ограниченно описываются в рамках ABI операционных систем, что требует осторожности при работе с ними. В рамках ABI, кажется (см. первую ссылку ниже почему «кажется»), стандартизировано представление таблицы виртуальных функций, за счёт чего можно взаимодействовать с DLL, собранными другим компилятором через абстрактные классы-интерфейсы, но это может влиять на производительность из-за виртуальности.


Ссылки:

1. Пространный пост по поводу ABI из блога Алёны C++ (ссылка).
2. О vtable в одном из вариантов ABI (ссылка).

В общем, есть некоторые проблемы.

 **semenyakinVS**  18 сентября 2016 в 20:14     0 

Но всё-таки, как мне кажется, для C характерно использование именно для низкоуровневого программирования под всякие микроконтроллеры и это основная область, где он живёт. Сужу по общению со знакомыми системщиками и по общим наблюдениям.

 **gnomod** 18 сентября 2016 в 21:06     -1 

Ну как минимум C более применим если у вас допустим 16к оперативной памяти и где-нибудь 32к для исполняемого кода, применять ассемблер это получить переносимый код, применять C++ тогда забудьте об операторах new т.к. ваша оперативная память на вес золота и вам необходимо только ручное управление памятью, кроме того обычно компилятор C дает более компактный и быстрый код

**VioletGiraffe** 18 сентября 2016 в 21:07

↑ +1 ↓

Идея понятна, хотя у меня не было проблемы писать на C++ с наследованием и виртуальными функциями под STM32 F1, там тоже памяти не мегабайты.

**gnemod** 18 сентября 2016 в 21:27

↑ 0 ↓

Просто нужно учитывать больше факторов при кодировании на C++ под мелкие контроллеры, например:

1. Фрагментация HEAP
2. Применение объектов в условиях жесткого реалтайма, когда все ресурсы контроллера расписаны по наносекундам (те вызов new и delete)
3. Контроль занимаемой памяти классами особенно с полиморфизмом и наследованием

вообщем появляется много новых нюансов, но если железо не высокотиражное и его ресурсы заложены с запасом то конечно и C++ годится

**robert\_ayrapetyan** 18 сентября 2016 в 22:06

↑ 0 ↓

Ядра всех ОС, движки топ 5 популярных БД, самые высокопроизводительные веб-серверы, движки практически всего 3д-софта, бытовая техника, начинка любого современного автомобиля, NASA...

Из этого следует, что на C писали (и пишут) самый надежный и безопасный код в мире.

**VioletGiraffe** 18 сентября 2016 в 22:24

↑ 0 ↓

М-м... нет, не следует. Это всего лишь ваше предположение.

**Yuuri** 19 сентября 2016 в 01:54

↑ +1 ↓

Пишут. Вопрос только, какой ценой, и сколько человекоэпох было отдано на отладку этих ядер и движков?

В перечисленных вами задачах в первую очередь важны скорость и/или близость к железу, в чём C действительно идеален, но это практически противоречит надёжности. Факторы надёжности — это, например, проверки операций (прощай, скорость), строгие системы типов (прощайте, низкоуровневые оптимизации), строгие спецификации (прощай, бесплатная переносимость на любой утюг). Ну правда, как можно назвать эталоном надёжности слаботипизированный язык, у которого стандарт состоит из UB чуть менее, чем наполовину?

**Alexeyslav** 20 сентября 2016 в 11:32

↑ -1 ↓

А кто и на чём будет писать низкоуровневые оптимизированные по быстродействию функции/библиотеки для высокоуровневых языков? Именно для этого и используется.

**VioletGiraffe** 20 сентября 2016 в 11:58

↑ +1 ↓

На ограниченном подмножестве C++, например.

**Yuuri** 17 сентября 2016 в 20:31

↑ +16 ↓

сочетает в себе надёжность C++

*нервный смешок*

Java был разработан Джеймсом Гослингом в 1990 году в компании Sun Microsystems.

Не знаю, кому верить, но Википедия говорит, что разработка только началась в 1991, а вышел на свет язык вообще в 1995.

Особенность Java заключается в том, что это первый чисто объектно-ориентированный язык программирования.

А это уже совсем неправда. Во-первых, Java не совсем чиста — как минимум, в ней есть примитивные типы, которые не являются объектами. Во-вторых, гораздо более расово чистый в этом отношении Smalltalk был уже в 80-х.

**BalinToms** 18 сентября 2016 в 14:54

↑ -8 ↓

@Yuuri сочетает в себе надёжность C++

— нервный смешок

на каких языках написаны 99% ОС и прикладноашем компэте ПО на вашем компьютере? Это и будет ответом на ваш смешок



**Arenim** 18 сентября 2016 в 15:19



↑ +5 ↓

> на каких языках написаны 99% ОС

Вообще-то на С, а не на С++

И да, я всякий раз с большим удовольствием цитирую Линуса Торвальдса, когда речь заходит про «С или С++»



**VioletGiraffe** 18 сентября 2016 в 16:38



↑ 0 ↓

ИМХО только потому, что во время написания тех ОС С++ ещё не был достаточно развит и допилен.



**codemax** 20 сентября 2016 в 16:22



↑ 0 ↓

Когда же он уже будет достаточно развит и допилен?.. Просто этого не случилось до сих пор, а комитет по стандартизации морозит и откладывает приятные новые фишки до тех времен, когда они уже совсем перестают казаться новыми и воспринимаются как нечто само собой разумеющееся.



**VioletGiraffe** 20 сентября 2016 в 16:29



↑ +1 ↓

Вы считаете, что не случилось, а я считаю, что с 2003 года язык полностью юзабелен, а с 2011 — прекрасен. На С++ 11 за вечер пишутся вещи, которые на С++03 я вообще не взялся бы писать.



**Arenim** 25 сентября 2016 в 00:05



↑ 0 ↓

интересно. А можно пример вещи, которая «пишется за вечер на с++11 и не пишется на с++03»?



**VioletGiraffe** 25 сентября 2016 в 00:24



↑ 0 ↓

1. Всё, где нужны лямбды и `std::function`.
2. Многопоточность. Здесь важен не столько класс `std::thread`, как те же самые лямбды и `function`.



**Arenim** 25 сентября 2016 в 01:00



↑ 0 ↓

гм. Вы сейчас сказали «лямбды — это круто». Никто собственно и не спорит, но примера-то и нету, так как «все, где нужны...» — не пример, как и слишком абстрактное «многопоточность» тож.



**VioletGiraffe** 25 сентября 2016 в 09:38



↑ 0 ↓

Лямбды — это не просто круто. Они позволяют делать то, что без них неоправданно сложно и в написании, и в применении.

Сигналы / слоты.

Пул потоков.

Блокирующая очередь, через которую N потоков последовательно сцепляются в конвейер.

См. Threading <https://github.com/VioletGiraffe/cpputils>



**Alexey2005** 17 сентября 2016 в 20:32



↑ +1 ↓

Удивительно, что Python настолько популярен. Трудно вообще найти такую нишу, в которой он был бы лучшим. Чтоб можно было сказать: да, вот эти вот задачи эффективнее всего делать на Python, другие языки заставят провозиться дольше и потратить больше ресурсов.

Практически всегда критерий его выбора субъективен, в стиле «конечно, на %language\_name% было бы быстрее и проще, но я его не знаю, да и синтаксис очень не нравится».



**keydon2** 17 сентября 2016 в 20:40



↑ 0 ↓

Затраты на обучение тоже затраты.  
Питон не знаю.



**Vlad\_IT** 17 сентября 2016 в 20:55



↑ +2 ↓

В парсинге данных, анализ данных, создание веб-приложений на коленке за короткое время, машинное обучение.



**MikeLP** 17 сентября 2016 в 21:02



↑ +6 ↓

Ну давайте, расскажите чем Python хуже PHP.  
Поверьте, звучать это будет одинаково:

Удивительно, что PHP настолько популярен. Трудно вообще найти такую нишу, в которой он был бы лучшим. Чтоб можно было сказать: да, вот эти вот задачи эффективнее всего делать на PHP, другие языки заставят провозиться дольше и потратить больше ресурсов.

или

Удивительно, что Ruby настолько популярен. Трудно вообще найти такую нишу, в которой он был бы лучшим. Чтоб можно было сказать: да, вот эти вот задачи эффективнее всего делать на Ruby, другие языки заставят провозиться дольше и потратить больше ресурсов.

и так далее...



**robert\_ayrapetyan** 17 сентября 2016 в 21:05



↑ +4 ↓

Python доминирует в нише бекенда и сервисов, где не требуется высокая производительность, поэтому его можно поставить в ряд с Ruby, PHP и JavaScript (SS). Любая попытка выбрать лучший из этих языков закончится «сравнением оппонента с Гитлером».

Я выбрал Python для себя потому, что он позволяет решать бизнес задачи, не думая ни о чем, кроме бизнес-логики, его синтаксис «чище» чем у вышеозначенных конкурентов, меньше всякой мишуры. В совокупности со знаниями C/C++ это позволяет реализовать систему/архитектуру любой сложности в короткие сроки и под любые нужды. Можно заменить слово Python на Ruby/PHP/JS(SS), но я выбрал именно Python, просто вот с первого знакомства, ни разу не возникло желания возвращаться на PHP/JS или пробовать Ruby. И да, «конечно, на %language\_name% было бы быстрее и проще» — это не про Python точно.



**ShashkovS** 17 сентября 2016 в 21:10



↑ +2 ↓

Ну, ок. Давайте ~~спорить~~ приводить примеры. Может быть, я просто сильно не в теме.  
Всё примеры моих реальных задач, которые я решаю питоном (но не стал бы даже браться за них, скажем, на плюсах).

Задача 1. есть html с задачами (оформленными единообразно) и excel-файл с их сложностями. Нужно добавить сложности в html (типа «Задача 13.7» -> «Задача 13.7 [\*\*\*]»);

Задача 2. прочитать данные из гугль-таблички, после чего разослать e-mail'ы в зависимости от данных там;

Задача 3. Есть TeX-файл с условиями задач. Сконвертировать его в html, используемые картинки сконвертировать в png, разложить по папкам, залить по FTP на сайт;

Задача 4. Есть excel с таблицей-плюсником (пустой). Есть отсканированный лист. Распознать плюсы и внести назад в excel;

Задача 5. Есть старая программа, работающая под DOS. Она генерит крупные отчёты достаточно сложного формата (руками делать заново сложно). Нужно сгенерить всевозможные отчёты, эмулируя действия пользователя;

Задача 6. Нужно пройти по пачке страниц в интернете и вытащить из них некую информацию.



**mezastel** 18 сентября 2016 в 00:32



↑ +3 ↓

На плюсах проблема с либами и их юзабельностью. А вот на C# все вышеописанные задачи решаются. Ну разве что распознать, но для этого и в Питоне нужна сторонняя либа.

НЛО прилетело и опубликовало эту надпись здесь



**equand** 19 сентября 2016 в 14:14



↑ +1 ↓

Зато пять лет спустя глядя на Ваш перл-скрипт Вы уйдете в запой... А питоновый скрипт написать для запоя надо умудриться.

НЛО прилетело и опубликовало эту надпись здесь



**maxim\_0\_o** 18 сентября 2016 в 20:26



↑ 0 ↓

Решить-то их можно на чем угодно. Вопрос в количестве потраченного времени и в объеме кода. И что-то мне подсказывает, что Java и C# уступят по этим критериям питону для этих задач. Тут следует обратить внимание, что в приведенных задачах не требуется слишком высокая производительность, что и служит знаком к использованию питона.



**Dark\_Daiver** 18 сентября 2016 в 07:42



↑ +1 ↓

Как минимум быстрое прототипирование в области scientific computing/machine learning/data mining, благо есть целая прорва либ (scipy/sympy/sklearn/pandas/theano/tensorflow/xgboost/opencv/etc) которые так или иначе увязаны вместе при помощи NumPy



**m1ld** 17 сентября 2016 в 20:59



↑ +2 ↓

Не совсем в тему, но на опубликованной странице также есть рейтинг самых популярных репозиторийев. Нашел для себя кладезь знаний <https://github.com/open-source-society/computer-science>.



**TWayne** 17 сентября 2016 в 21:03



↑ 0 ↓

Действительно, предсказуемый рейтинг



**lenar** 17 сентября 2016 в 21:11



↑ +2 ↓

Автор не в курсе, что JS — это уже далеко не только фронтендовый язык?



**Trueteller** 17 сентября 2016 в 21:21



↑ +3 ↓

Характеристики языков — огонь.

Многие тезисы как минимум спорны, стилистика описания разных языков сильно различается, да и еще и пунктуация хромает.

| JavaScript, C# и Go показали двойной рост аудитории. А аудитория Swift и TypeScript выросла в 3,5 раза

Это в сравнении с каким периодом? Как мог C# вырасти в 2 раза за последний год-два?



**zapolnoch** 17 сентября 2016 в 21:39



↑ +3 ↓

Может за счет Xamarin?



**Antelle** 17 сентября 2016 в 21:41



↑ +3 ↓

Microsoft же пилят кроссплатформенный C# последний год-два, конечно % растёт.



**m1ld** 17 сентября 2016 в 21:57



↑ 0 ↓

Если вы об этом <https://blogs.msdn.microsoft.com/dotnet/2016/04/01/net-at-build-2016-open-cross-platform-and-free/>? То меня дата публикации как-то настораживает.



**Antelle** 17 сентября 2016 в 22:09

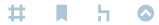


↑ +2 ↓

Xamarin ещё. В целом C# становится приятнее на разных платформах, наверное поэтому его больше появляется на github. Напомню, что статистика именно по коду на гитхабе, а не вообще о языке.



**Trueteller** 17 сентября 2016 в 22:20



↑ +1 ↓

Да, скорее всего это именно за счет кода, выложенного самим Microsoft на github. Ну и того толчка, что это дало C# open source.



**Antelle** 17 сентября 2016 в 22:26



↑ +1 ↓

Ага. Если посмотреть в тренды по C#: powershell, roslyn, kestrel, .net core, bot builder — появились и сами проекты, и всякое разное для них, ms стали активно работать с продвижением языка в open source (что не может не радовать).



**KirillFormado** 20 сентября 2016 в 16:24



↑ 0 ↓

На вскидку еще могут быть open-source игры на Unity ну и моды для игр. Из того что знаю, много модов для Kerbal Space Program лежит на github-e



**VaalKIA** 17 сентября 2016 в 21:52



↑ +1 ↓

А нет метрики: «количество новых проектов» на каком-то языке? Мне кажется, именно это было бы достоверным показателем популярности.



**shalaev** 17 сентября 2016 в 21:57



↑ 0 ↓

некоторые весьма известные и надежные сайты работают на python, особенно pinterest.com, instagram.com и rdio.com последний — особенно надежный



**VGrabko** 18 сентября 2016 в 00:00



↑ +4 ↓

У Go нет интерпретатора. go run просто компилирует во временную директорию и запускает готовый бинарник



**asvechkar** 18 сентября 2016 в 00:29



↑ 0 ↓

Небольшое дополнение к п.6 о создателе языка C++: <https://geektimes.ru/post/77210/>



**Error1024** 18 сентября 2016 в 00:33



↑ +10 ↓

В статье куча странных мест:

Кроме того, некоторые современные игры и операционные системы были разработаны на C++ из-за быстрого процессинга и компиляции.

Это у C++ компиляция быстрая???

Язык C# позволяет разрабатывать практически любые приложения, которые связаны с Visual Studio IDE.

Эээ, это вообще про что?

JavaScript был разработан компанией Netscape и вряд ли есть сайты, которые не используют его.

Есть куча таких сайтов.



**sugadu** 20 сентября 2016 в 16:21



↑ 0 ↓

Есть куча таких сайтов.



Дайте, пожалуйста, ссылки на самые популярные, интересно глянуть.



**rie** 18 сентября 2016 в 02:41



-7



Не могу дождаться, когда мыльный пузырь веба взорвется и похоронит сам себя, а так же потянет за собой все эти пыхопе, джэс, раби. То что твориться сейчас в вебе это не помойка даже, это параша. Деградация и нивелирование всех постулатов программирования.

Люди пишут полнейший говнокод выдавая это за ИННОВАЦИЮ и НОВЫЕ ПАРАДИГМЫ, каждый день делая ЕЩЕ ОДНУ ПРИНЦИПИАЛЬНО НУВУЮ либу для веба, и пользуются в своих проектах таким же говном.

Вся суть.



**ankh1989** 18 сентября 2016 в 11:34



0



Насчёт мыльного пузыря не знаю, но "твориться" пишется без "ь".



**spyderDFX** 20 сентября 2016 в 09:10



0



Руби.

P.S. Haters gonna hate



**DevMan** 18 сентября 2016 в 13:10



+8

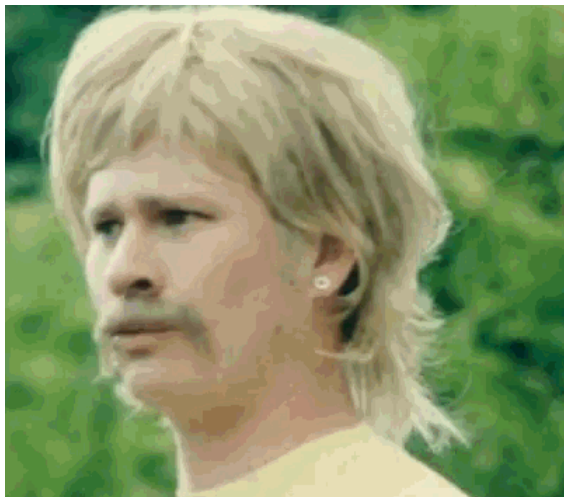


Он был создан для iPhone-приложений

Ruby — простой и читаемый язык программирования, ориентированный на разработку веб приложений

фреймворк Ruby использовался для разработки Github, Scribd, Yammer, Shopify и Groupon

Особенность Java заключается в том, что это первый чисто объектно-ориентированный язык программирования и множество других перлов.



**tmn4jq** 18 сентября 2016 в 18:14



+2



Pearl'ов

Ruby сочетает в себе некоторые возможности Lisp, Pearl и Eiffel



**loz** 20 сентября 2016 в 01:57





0



>Особенность Java заключается в том, что это первый чисто объектно-ориентированный язык программирования

Ахаха, спасибо, понравилось описание языков. Захожу на хабр и сразу вспоминаю почему я сюда не захожу)

 **thims** 20 сентября 2016 в 16:20  

 +1 

Язык C# позволяет разрабатывать практически любые приложения, которые связаны с Visual Studio IDE.


Это как понимать??

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

- Сутки
- Неделя
- Месяц

Тест старения светодиодных ламп

 +126  31k  49  149

Хватит натягивать сову на глобус

 +31  18,1k  35  93


Тёмная сторона работы в Яндекс.Маркете

 +147  17,9k  52  122

Советы по выбору усилителя сигнала сотовой связи 2G/3G/4G/5G

 +24  13,7k  85  21

IT-ассоциации предупредили власти РФ об опасности массового отъезда программистов за рубеж

 +24  13k  10  72

<b>Ваш аккаунт</b> Войти	<b>Разделы</b> Публикации	<b>Информация</b> Устройство сайта	<b>Услуги</b> Реклама
Регистрация	Новости	Для авторов	Тарифы
	Хабы	Для компаний	Контент
	Компании	Документы	Семинары
	Пользователи	Соглашение	Мегaproекты
	Песочница	Конфиденциальность	

Если нашли опечатку в посте, выделите ее и нажмите Ctrl+Enter, чтобы сообщить автору.