

The Incredible Growth of Python



by [David Robinson](#) on September 6, 2017

We [recently explored](#) how wealthy countries (those defined as [high-income](#) by the World Bank) tend to visit a different set of technologies than the rest of the world. Among the largest differences we saw was in the programming language Python. When we focus on high-income countries, the growth of Python is even larger than it might appear from tools like [Stack Overflow Trends](#), or in other rankings that consider global software development.

In this post, we'll explore the extraordinary growth of the Python programming language in the last five years, as seen by Stack Overflow traffic within high-income countries. The term "fastest-growing" can be [hard to define precisely](#), but we make the case that **Python has a solid claim to being the fastest-growing major programming language**.

All the numbers discussed in this post are for **high-income countries**; they're generally representative of trends in the United States, United Kingdom, Germany, Canada, and other such countries, which in combination make up about 64% of Stack Overflow's traffic. Many other countries such as India, Brazil, Russia, and China also make enormous contributions to the global software development ecosystem, and this post is less descriptive of those economies, though we'll see that Python has shown growth there as well.

It's worth emphasizing up front that the number of users of a language isn't a measure of the language's quality: we're *describing* the languages developers use, but not prescribing anything. (Full disclosure: I [used to program](#) primarily in Python, though I have since switched entirely to R).

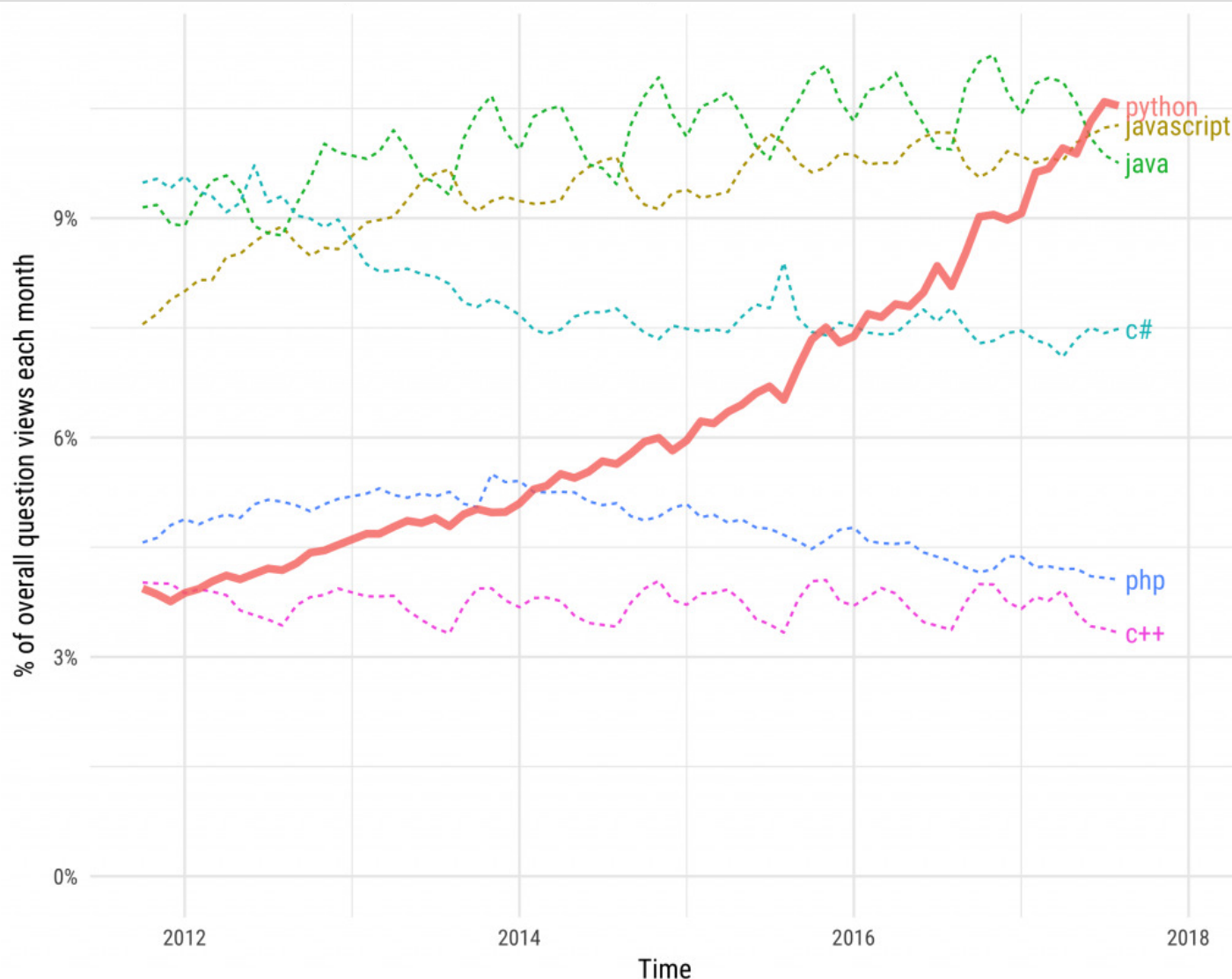
Python's growth in high-income countries

You can see on [Stack Overflow Trends](#) that Python has been growing rapidly in the last few years. But for this post we'll focus on high-income countries, and consider visits to questions rather than questions asked (this tends to give similar results, but has less month-by-month noise, especially for smaller tags).

We have data on Stack Overflow question views going back to late 2011, and in this time period we can consider the growth of Python relative to five other major programming languages. (Note that this is therefore a shorter time scale than the Trends tool, which goes back to 2008). These are currently six of the ten most-visited Stack Overflow tags in high-income countries; the four we didn't include are CSS, HTML, Android, and JQuery.

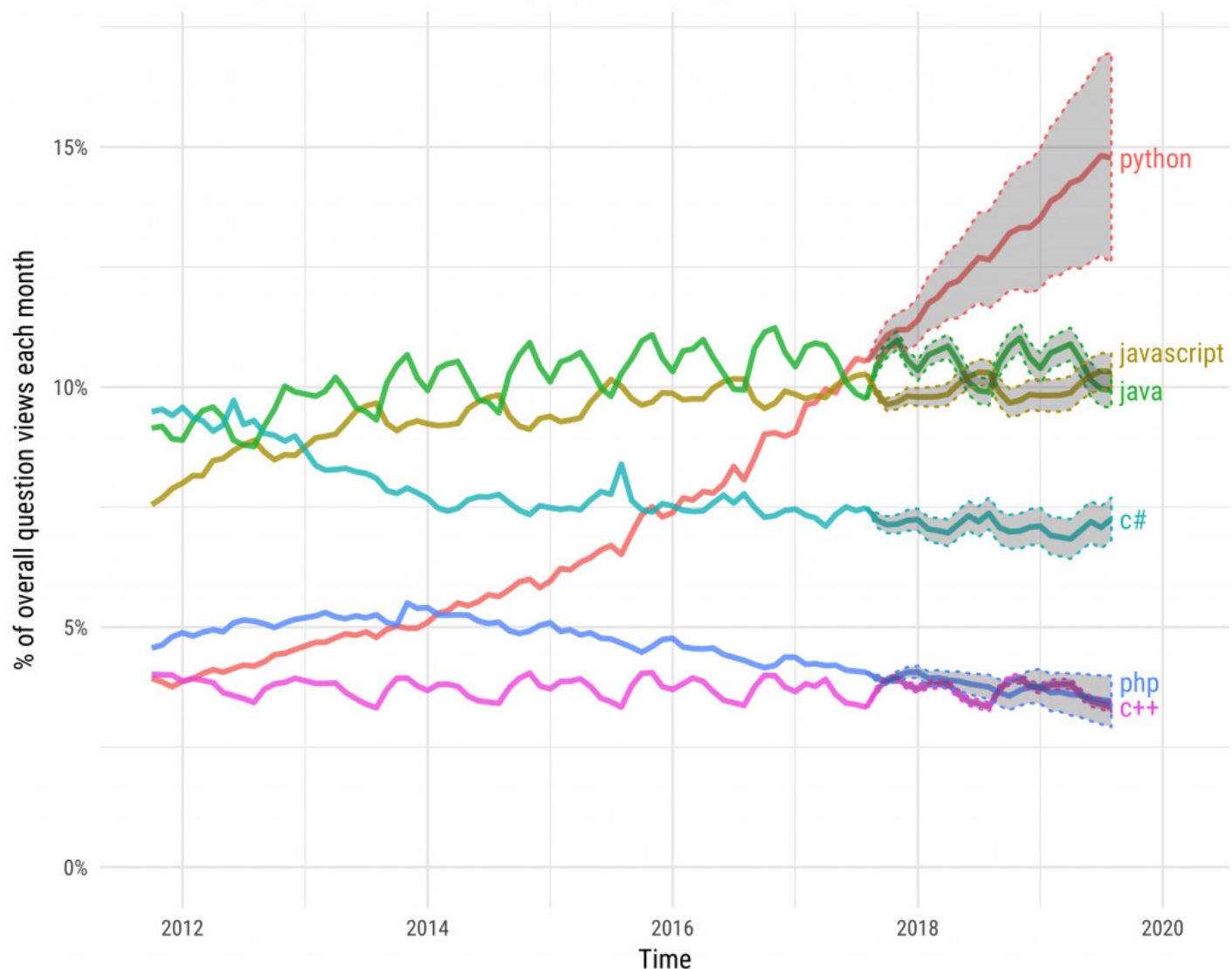


4.3K



June 2017 was the first month that **Python was the most visited tag on Stack Overflow within high-income nations**. This included being the most visited tag within the US and the UK, and in the top 2 in almost all other high income nations (next to either Java or JavaScript). This is especially impressive because in 2012, it was less visited than any of the other 5 languages, and has grown by 2.5-fold in that time.

Part of this is because of the seasonal nature of traffic to Java. Since it's [heavily taught in undergraduate courses](#), Java traffic tends to rise during the fall and spring and drop during the summer. Will it catch up with Python again by the end of the year? We can try forecasting the next two years of growth with a [model called "STL"](#), which combines growth with seasonal trends to make a prediction about future values.



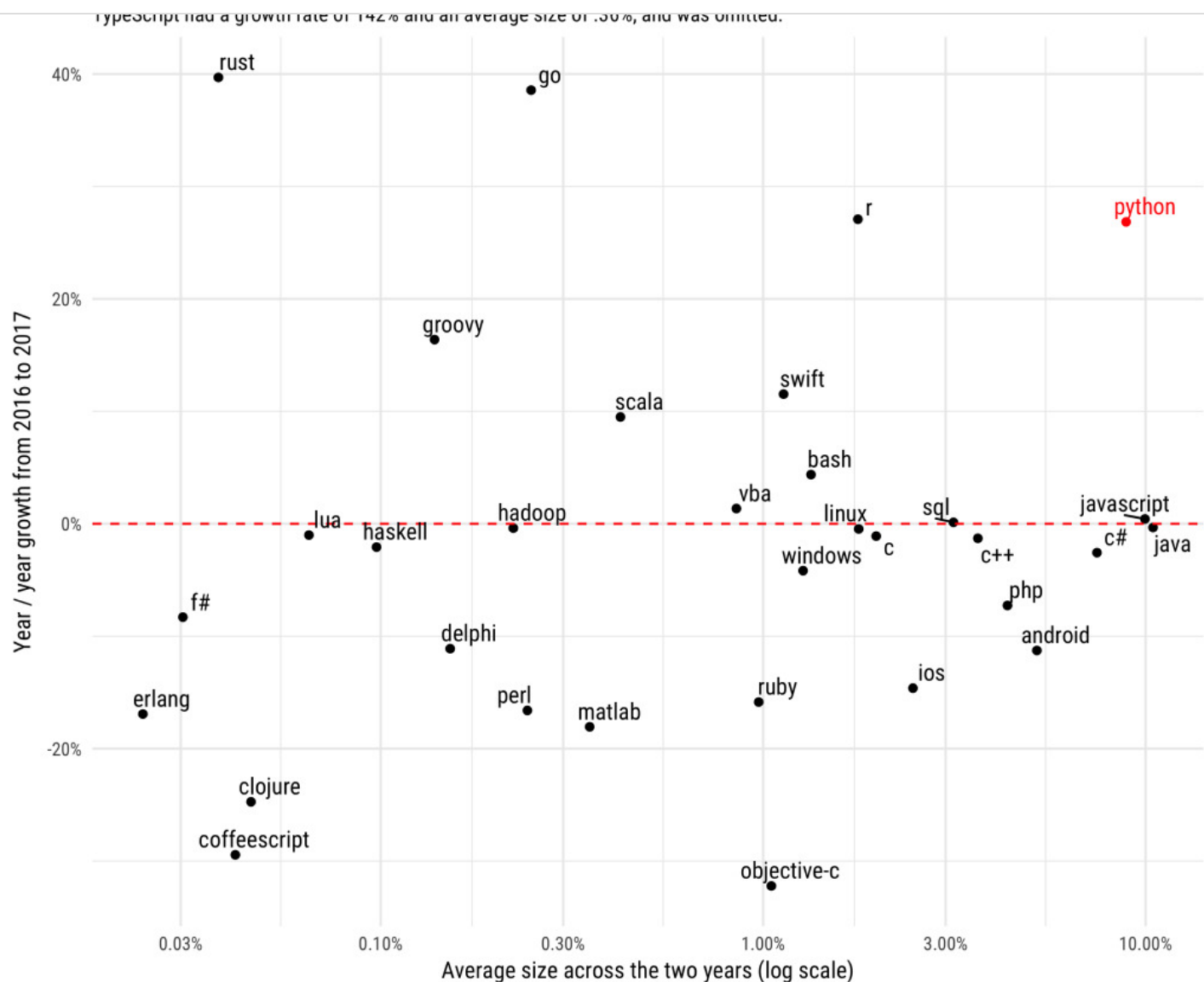
According to this model, Python could either stay in the lead or be overtaken by Java in the fall (it's roughly within the variation of the model's predictions), but it's clearly on track to become the most visited tag in 2018. STL also suggests that JavaScript and Java will remain at similar levels of traffic among high income countries, just as they have for the last two years.

What tags are growing the fastest overall?

The above was looking only at the six most-visited programming languages. Among other notable technologies, which are currently growing the fastest in high-income countries?

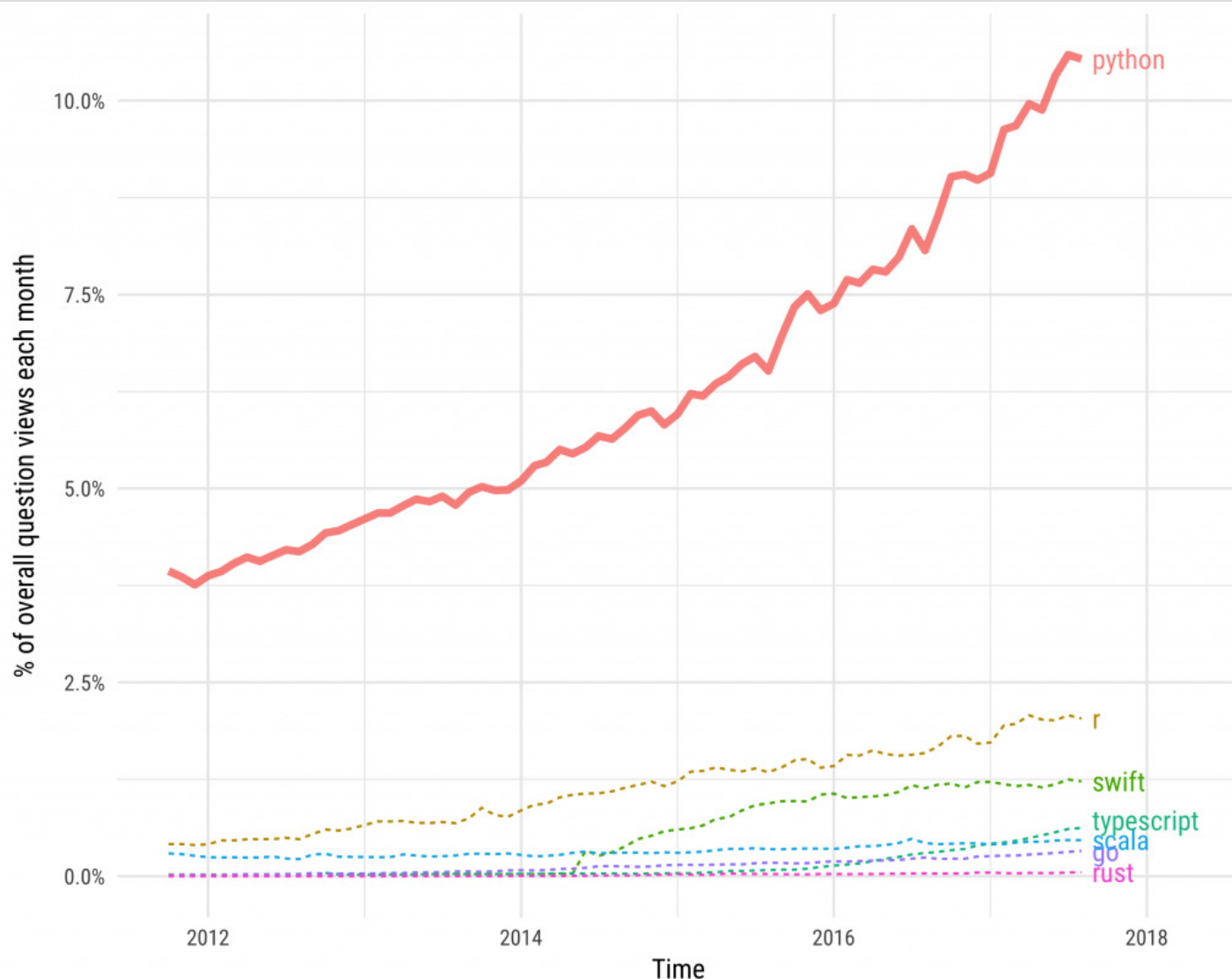
We defined the growth rate in terms of the ratio between 2017 and 2016 share of traffic. We decided to consider only programming languages (like Java and Python) and platforms (such as iOS, Android, Windows and Linux) in this analysis, as opposed to frameworks like [Angular](#) or libraries like [TensorFlow](#) (although many of those showed notable growth that may be examined in a future post).

Because of the challenges in defining "fastest-growing" described in [this comic](#), we compare the growth to the overall average in a [mean-difference plot](#).



With a 27% year-over year-growth rate, Python stands alone as a tag that is **both large and growing rapidly**; the next-largest tag that shows similar growth is R. We see that traffic to most other large tags has stayed pretty steady within high-income countries, with visits to Android, iOS, and PHP decreasing slightly. We previously examined some of the shrinking tags like Objective-C, Perl and Ruby in our [post on the death of Flash](#)). We can also notice that among functional programming languages, Scala is the largest and growing, while F# and Clojure are smaller and shrinking, with Haskell in between and remaining steady.

There's an important omission from the above chart: traffic to TypeScript questions grew by an impressive 142% in the last year, enough that we left it off to avoid overwhelming the rest of the scale. You can also see that some other smaller languages are growing similarly or faster than Python (like R, Go and Rust), and there are a number of tags like Swift and Scala that are also showing impressive growth. How does their traffic over time compare to Python's?



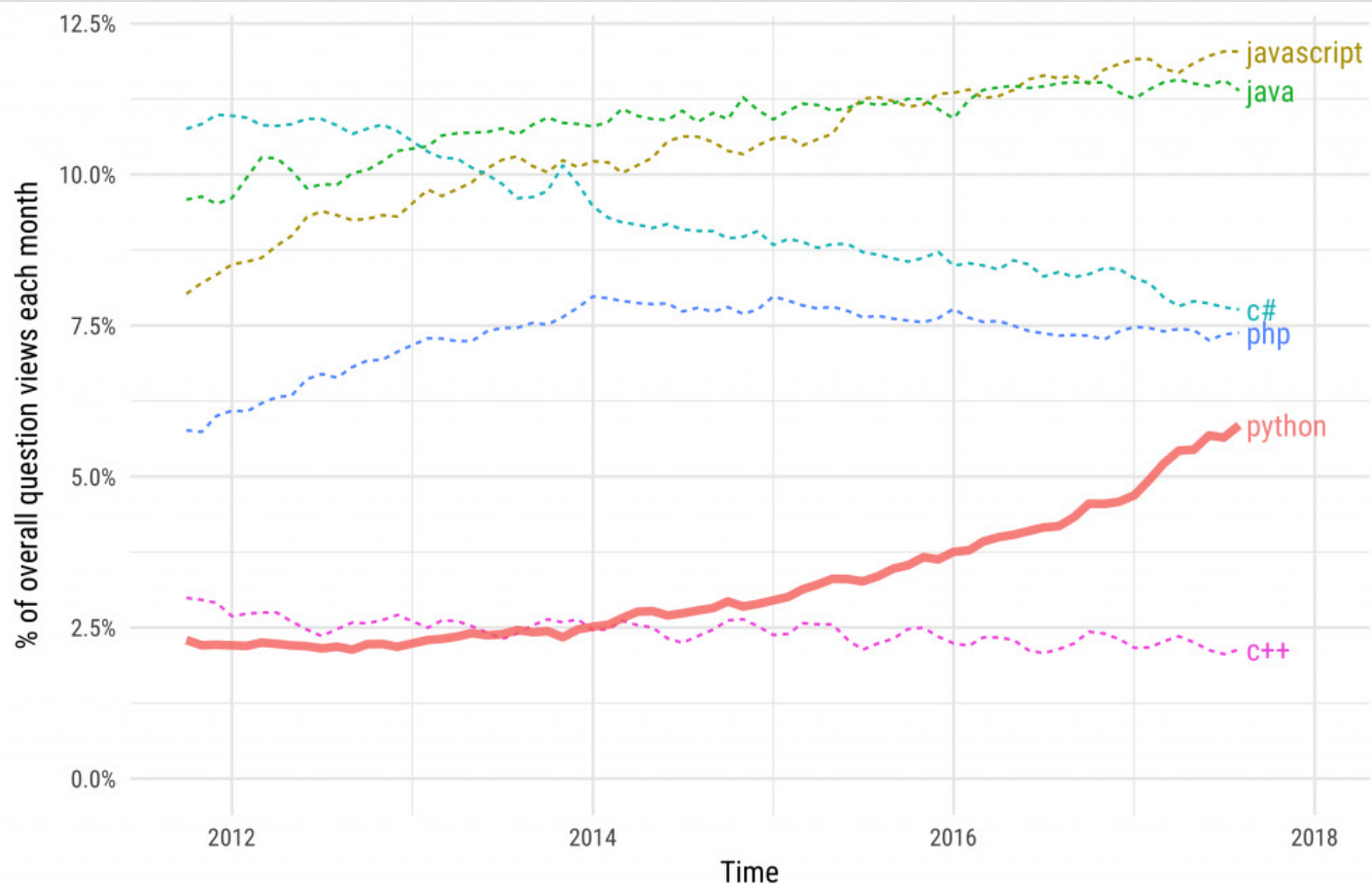
The growth of languages like R and Swift is indeed impressive, and TypeScript has shown especially rapid expansion in an even shorter time. Many of these smaller languages grew from getting almost no question traffic to become notable presences in the software ecosystem. But as this graph shows, it's easier to show rapid growth when a tag started relatively small.

Note that we're not saying these languages are in any way "competing" with Python. Rather, we're explaining why we'd treat their growth in a separate category; these were lower-traffic tags to start with. Python is an unusual case for being **both one of the most visited tags on Stack Overflow and one of the fastest-growing ones**. (Incidentally, it is also accelerating! Its year-over-year growth has become faster each year since 2013).

Rest of the world

So far in this post we've been analyzing the trends in high-income countries. Does Python show a similar growth in the rest of the world, in countries like India, Brazil, Russia and China?

Indeed it does.



Outside of high-income countries Python is *still* the fastest growing major programming language; it simply started at a lower level and the growth began two years later (in 2014 rather than 2012). In fact, the year-over-year growth rate of Python in non-high-income countries is slightly *higher* than it is in high-income countries. We don't examine it here, but R, the [other language whose usage is positively correlated with GDP](#), is growing in these countries as well.

Many of the conclusions in this post about the growth and decline of tags (as opposed to the absolute rankings) in high-income countries hold true for the rest of the world; there's a 0.979 Spearman correlation between the growth rates in the two segments. In some cases, you can see a "lagging" phenomenon similar to what happened with Python, where a technology was widely adopted within high-income countries a year or two before it expanded in the rest of the world. (This is an interesting phenomenon and may be the subject of a future blog post!)

Next time

We're not looking to contribute to any "language war." The number of users of a language doesn't imply anything about its quality, and certainly can't tell you which language is [more appropriate for a particular situation](#). With that perspective in mind, however, we believe it's worth understanding what languages make up the developer ecosystem, and how that ecosystem might be changing.

This post demonstrated that Python has shown a surprising growth in the last five years, especially within high-income countries. In our next post, we'll start to explore the "*why*". We'll segment the growth by country and by industry, and examine what other technologies tend to be used alongside Python (to estimate, for example, how much of the growth has been due to increased usage of Python for web development versus for data science).

Author



David Robinson

Data Scientist

[Author Archives](#) • [Website](#) • [Twitter](#) • [Github](#)

Tags

[announcements](#)

[insights](#)

Related Articles

Stack Overflow for Teams is Now Available

by Alex Miller on May 3, 2018

It's here. After months of designing, building, testing, and beta-ing, we're happy to announce that as of today, Stack Overflow for Teams has launched and is available to everyone!

[Continue reading](#)

<https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

Stack Overflow Isn't Change.

by Jay Hanlon on April 26, 2018

Let's start with the painful truth:

Comments

**thecity2**

September 6, 2017 at 1:44 pm

The “why” for Python (and to a large extent R) is fairly obvious: The rise of Data Science/ML/Deep Learning. Deep Learning frameworks available with Python APIs, in addition to Spark for “Big Data”, combined with the ease of picking up Python and a couple decades worth of scientific packages coming from academia and industry have made Python an incredibly productive and versatile Swiss Army Knife.

**Herr Derb**

September 6, 2017 at 4:30 pm

Swiss Army Knife for individual science and research projects. I really would like to know for what kind of projects these programming languages are used. I simply cannot imagine python being used for a regular industrial software project.

**Max Malysh**

September 6, 2017 at 5:11 pm

The first version of Google was written in Python.

**discuss**

September 12, 2017 at 4:44 pm

Very true. Best advice I got early in career listening to a Google Engineer was to in fact learn new languages as they come out. not just ones employers looking for at the moment.

**jgmitzen**

September 6, 2017 at 6:56 pm

It is indeed used for regular software projects:

<https://www.python.org/about/success/>

From Lucasfilm to Paypal (where it's used in over 50 projects)...

<https://www.paypal-engineering.com/2014/12/10/10-myths-of-enterprise-python/>

...to Netflix:

<https://medium.com/netflix-techblog/python-at-netflix-86b6028b3b3e>

**Alex Punnen**

September 6, 2017 at 11:25 pm

It is indeed used, I guess this is used mostly by start ups to implement an idea; and then it grows and grows; and is handed over to commercial programmers who have to deal with the nightmare of maintenance and hard refactoring.

I am not sure you could call Google, Facebook and Youtube "startups"

And large opensource projects with thousands of developers don't seem to have a problem maintaining python. (openstack, ansible)

I think the issue is probably with the culture or practices at the places you have worked more than any language they chose to use.

Here is a good example of a project that refactored java to python to simplify dependencies and coding.

https://bugs.documentfoundation.org/show_bug.cgi?id=38820

Tensorflow and pytorch have lots of traction in the Machine Learning world too. ML/DL/etc... have added to the popularity too.

But to be brutally honest, most ugly python programs are only ugly because the developers didn't take the time to learn the idiomatic, and pythonic way of using the language. But once again that is not the languages fault, that is the fault of developers who cargo culted bad practices from other languages.



Ravan Asteris

September 12, 2017 at 2:03 pm

My first experience with Python was trying to make sense of Anaconda (the red hat distro packaging software) in 2002.

Horrible, uncommented spaghetti! Space delimited crap! Unreadable, not enough white space (see space delimited), zero meaningful comments, few comments at all. Utter garbage. I'm still shocked that it ran.



Alex Punnen

September 6, 2017 at 11:23 pm

Agree, I have used Python in one project and though I love the simplicity in prototyping with it, the dynamic typing part makes it extremely hard to maintain especially as more and more teams work on it, and when the project is 2 , 3 years old.



Giorgi Moniava

September 7, 2017 at 6:06 am

Yeah I would also like to know what are the strengths of this language over languages such as C#, JS, or Java.



new year

September 24, 2017 at 3:28 am

Stack Overflow traffic within high-income countries. The term "fastest-growing" can be hard to define precisely, but we make the case that Python has a solid claim to being the fastest-growing major programming language.



Vince O'Sullivan

September 24, 2017 at 5:13 am

I'd really like to see that case made (using much better statistics than "questions asked on overflow"). One alternative measure is the Tiobe Index. You may not agree with the way it measures a language's popularity but has been a reasonably consistent measure taken for the last 15 years. It doesn't contain any hint that "Python has a solid claim to being the fastest-growing major programming language".

<https://www.tiobe.com/tiobe-index/>



camden_kid

September 6, 2017 at 2:55 pm

**Tadeu Manoel**

September 6, 2017 at 3:03 pm

Where is Julia? 😊

**David Robinson**

September 6, 2017 at 3:26 pm

<https://uploads.disquscdn.com/images/c98fb169ba110c6bd1c81d0865ad13eb420e138ba753343dcb30e06c3a6550aa.png>

**aNoobNoob**

September 6, 2017 at 3:41 pm

What about Elixir 😊

**Willy Karam**

September 6, 2017 at 10:57 pm

@David Robinson I would also echo that question: "What about Elixir?"

**David Robinson**

September 7, 2017 at 4:39 pm

Elixir would have looked very similar to Rust- you're probably right that it could have been included! (May even have the chance to edit this graph to add it). At that size there's a question of where to draw the line (I didn't bother including, say, Fortran) but admittedly it's a bit unfair to mention Rust but not Elixir.

<https://uploads.disquscdn.com/images/3593321bf514836d91cc43842b035a2f03ea2bc09373270076ccc3460294b9d9.png>

**Willy Karam**

September 7, 2017 at 5:06 pm

Thanks very much for the update. I'd say elixir is on a trajectory that's more similar to Go, just not as far along in the cycle. Given that it's later release than Rust/Go, it'll be interesting to see these numbers over the next year or two — particularly on the year over year plot chart. Thanks very much for your work on the post and all the replies.

**Tadeu Manoel**

September 6, 2017 at 4:26 pm

Yiiikes.. it's still... not even...

**Willy Karam**

September 6, 2017 at 10:57 pm

Where is Elixir? 😊

**NewWorld**

September 7, 2017 at 7:53 am

It's there, but the x-axis overlapped it perfectly so you cannot see it.

**Bill Sommers**

September 7, 2017 at 4:35 pm

**Jerry Wasinger**

September 6, 2017 at 3:19 pm

Do you suppose that this correlates with the sales of the Raspberry Pi? I didn't know a thing about Python until I started programming my first RPi. Now it's my go-to language at pi-plates.com.

**Izkata**

September 6, 2017 at 4:02 pm

The first release of Raspberry PI (Feb 2012) correlates perfectly with when the growth began in the wider graph [in David Robinson's comment below](#)

**Kenneth J Hughes**

September 6, 2017 at 4:54 pm

See also Hacker News discussion: <https://news.ycombinator.com/item?id=15186025>

**Alec**

September 6, 2017 at 4:56 pm

Newbies to programming may be more likely to search for or ask more questions than experienced programmers.

Could this perhaps indicate that people new to programming often learn Python as a first language, and thus ask more questions about it? That is, could it be that Python's apparent popularity is magnified because its users are more likely to ask questions than other languages' users?

**Mike Croft**

September 6, 2017 at 5:04 pm

There is also frameworks to consider. For example, a user of Java EE is clearly using Java, but will not necessarily include the `java` tag in their question. I would assume the same is true of other languages, particularly Javascript.

**Bryan Oakley**

September 6, 2017 at 5:59 pm

You can say the same about python, too. A quick check shows me 74,000+ questions tagged with django but not with python. That number is a whole lot bigger than I expected it to be.

**Manthan Thakar**

September 6, 2017 at 5:05 pm

That's a good point. But I think it might also have something to do with a ton of things that you can do with Python. From building web apps, mobile apps to networking apps and now data-based applications/prototypes. It might just be that Python makes it easier for people to switch roles and try new things.

**Mark Hancock**

September 6, 2017 at 10:26 pm

It may also be that Python Code is very difficult to maintain (my experience)

**Ravan Asteris**

September 12, 2017 at 1:59 pm

**js**

September 7, 2017 at 9:39 am

For anyone wondering, yes there are many types of statistical bias and not taking these into account can lead you to bad conclusions.

**sth**

September 7, 2017 at 12:33 pm

If previously there were just a few newbies asking question and now there are more and more newbies asking questions, it means the language is growing.

The point is that the number of questions asked is increasing. So either the existing Python users are asking more and more questions as they gain experience, or new Python users are showing up.

**Alex P**

September 6, 2017 at 5:08 pm

Can we use an STL model on percentages like that? I'm thinking no. The timeseries are correlated – for example, the author explains that java is heavily taught in undergraduate courses, causing peaks in its share of views during the fall and spring college semesters. If you look at javascript, it dips when java rises and rises when java dips. Is that because all the college students read up on javascript during their breaks? more likely, the absolute page views of javascript are less correlated to undergraduate schedules, but its relative rank increases as the page views of java decline.

**microamp**

September 6, 2017 at 6:42 pm

I'd also like to see how Ruby does vs Python and others in Japan. And Vue.js vs React and Angular in China. That would be interesting data to show IMO.

**merito**

September 6, 2017 at 10:29 pm

I don't think we can get information about China users, they must pass the great fire wall :)

**wonho ha**

September 6, 2017 at 6:59 pm

i don't know the "language war". what is the "language war"?

**asd**

September 7, 2017 at 9:18 pm

like Nsync vs. backstreet years ago

**Nick Feller**

September 6, 2017 at 7:33 pm

Yo the java pattern is sooo funny. Slumps during summer and Christmas. These people are not real programmers, schools and white collar 😊

**Dr Porostertartamunduskusz**

September 8, 2017 at 7:31 am

**Sillas Gonzaga**

September 6, 2017 at 9:25 pm

Is the code for this analysis available?

**Naranmandakh Tsogoo**

September 7, 2017 at 1:09 am

i thought it is not actually growth , maybe it caused python's documentation poor less than java

**Thomas Güttler**

September 7, 2017 at 1:29 am

Great survey. It would be nice to see a survey like this for databases. At least in my daily business I see a switch of the focus. Clear and simple data structures are the base for clear and simple (Python) code.

**Ryan Graham**

September 7, 2017 at 1:59 am

I'm a J2EE developer by trade, a python developer on the side —
read: right tool for the right job. Maybe due to industry trends? Python seems to attract more academics — mathematicians, data scientists, physicists, etc. It's more approachable. Recent trends in AI and data science might suggest the libraries supporting these hot topics developed by such people make it more approachable by practitioners or experimenters. Who knows? Good read.

**Cpt Vineum**

September 7, 2017 at 4:14 am

Time to reopen my Python books and start to play with it!

**smitty**

September 7, 2017 at 11:39 am

Start with the latest version. Important new goings-on.

**HRIDAYESH SHARMA**

September 7, 2017 at 4:51 am

I learnt C++ and Python and wanted to take both of them at proficient level but I was confused for which one should i pick first. Finally I chose python.

**FDominicus**

September 7, 2017 at 4:58 am

It may grow as much as it wants, I don't like Python that much. I get along, but I do not embrace it.

**AI T**

September 7, 2017 at 5:10 am

That typescript projection is way off. See angular.



more to the project than my previous

**Vivek Jain**

September 7, 2017 at 5:16 am

Does JavaScript include other frameworks like NodeJs or AngularJs? Same for Java, does it include Android? And PHP? I hope there would be some overlap, even though it is marginal. Thoughts?

**js**

September 7, 2017 at 9:31 am

My guess is that it's specifically the language tags. So [Android] [Java] would count for both (but android is not a language per say, so it's not listed here). For javascript, [node.js][javascript] would be included but [node.js] only would not.

**amz3**

September 7, 2017 at 5:39 am

Looking forward the next post!

**neelkadia**

September 7, 2017 at 5:46 am

AI and ML are coming bro!

**Giorgi Moniava**

September 7, 2017 at 6:03 am

too many languages and too many frameworks this sounds as negative more to me than a positive

**aCppDev**

September 7, 2017 at 6:34 am

Very interesting graphs. For myself, the most interesting details are the yearly changes and emerging patterns especially for java, javascript and c++. My interpretation, java and c++ are the main languages when it comes to employees with daily 9 to 5 job, they decrease for the typical vacations around summer and new year. In contrast to this javascript seems to be the side-project/personal-use language, it rises when developers go on vacation. 😊 And php seems to be a christmas language maybe all the good old shopping sites need an update just before the big sales. So conclusion for me atleast, your best bet for a steady job with good work life balance are still java and c++.

**Iron Sean**

September 7, 2017 at 3:13 pm

I think students taking programming courses in Java and C++ (still the most common intro to programming languages) might explain that even more directly.

**Toby**

September 7, 2017 at 6:45 am

Interesting to note the very regular periodicity of Java and C++ in the first graph. Does this indicate their maturity as languages and in their use?

As well languages well established with a history of reliability and so on they are likely used as the main 'workhorse' languages of larger established, companies with western operations (say, Siemens for e.g.). This might correlate to the periodic pattern where work in such companies generally tends to slow somewhat over the Christmas and

**Philip Bayer**

September 7, 2017 at 10:04 am

Java is also an extremely typical language for college intro to programming classes. So stack overflow likely is flooded with Java question visits at the beginning of each semester and fewer in the summer

**BlueRaja**

September 7, 2017 at 7:43 pm

I'd guess it has more to do with those being commonly taught languages in schools. The drops seem to coincide with summer/spring/winter/fall vacations.

**Ettore Rizza**

September 7, 2017 at 7:55 am

“we’re describing the languages developers use ”

Is it not rather the languages for which people ask themselves many questions and on which they seek an answer on SO? As a general language widely used by beginners, sometimes in areas far away from computer science, it seems normal that Python attracts a lot of attention. I would be interested to know which libraries of Python are concerned. The hype around data science and machine learning has undoubtedly increased the number of questions related to modules such as Pandas, TensorFlow or Scikitlearn.

**Lazza**

September 8, 2017 at 7:28 pm

You have a great point. SO wrote “developers” while in reality they are considering programmers. A lot of Python programmers are not even developers or computer scientists (but this is also true for R and several great languages commonly used for data science, statistics, biology and a lot of other purposes).

**yanggeorge**

September 7, 2017 at 9:47 am

R is bad.

**215756347@qq.com**

September 7, 2017 at 9:16 pm

Please switch to Python. So bad nobody in company pushes me to Python so I stay in R with comfortably.

**Another Pt**

September 7, 2017 at 9:57 am

This is my 20th year anniversary of taking up Python! Hail Guido.
R growth is interesting – as this is free replacement for S+. Learned this back in 1990's as S.
I dumped S+ & could not stand Java or C++ 20 years ago.
I still love it but beginning to learning Scala nowadays. It won't replace Python anytime soon but a nice complement Python for speed & multi threading issues. Scala is great, but not quite as fun or easy as Python.
Hopefully, Scala won't take 20 yrs for me to monetize (does not seem like it either).

**smitty**

September 7, 2017 at 11:38 am

Guido is a BeauDiFuL man.

Not abusing his authority as benevolent dictator either unlike some. Beautiful mind. He is taking up issues with autistic children. He deserves to be supported.

**James Howe**

September 7, 2017 at 11:41 am

> Python for speed & multi threading issues.
Yet most Python implementations don't support multithreading.

**Richard Rast**

September 7, 2017 at 12:09 pm

Right, that's why s/he is using scala for that.

**Steve**

September 7, 2017 at 10:58 am

Could it be that folks flock to StackOverflow because they can't figure it out on their own and it's not well documented?

**sth**

September 7, 2017 at 12:02 pm

That would only explain the steady increase if you assume the Python documentation gets worse and worse each month.

**Paula**

September 8, 2017 at 1:03 am

I agree that the documentation is not Python's forte, but have you ever contributed to improving it?

**alanjds**

September 7, 2017 at 12:50 pm

Could. But specially not likely within Python. It have a community and design focus towards good documentation.

**P.B. Lecavalier**

September 7, 2017 at 1:04 pm

Official documentation is good but often lacks practical examples on how to make that thing dance. Also, I plead guilty that when using a module I never used before, I often wonder how to it in a pythonic way. StackOverflow answers that neatly. If it were another language, I would care less (and good luck to the future maintainer). That might explain why Java jobs pay more on average, according to some sources.

**Ravan Asteris**

September 12, 2017 at 1:53 pm

Hahahahaha.

No. The docs I've read are sub-par, and poorly indexed. I see better from third parties.



**John Rosewicz**

September 7, 2017 at 1:41 pm

I doubt this is a primary factor, but could definitely be a cause for some of the growth.

Python is a good language for beginners. Beginning programmers often have trouble with documentation, and, in my experience, Python's documentation is not particularly friendly. With more and more people learning to program, there are more beginner programmers than ever right now.

**Saelyth**

September 7, 2017 at 2:56 pm

You are correct. I am still learning and I check the documentation often when I want to use something new. The documentation helps me to understand how it works "more or less" but without a practical example I cannot make it work, most of the times. Stackoverflow has so many examples that a quick search fits my needs.

**Roger Deloy Pack**

September 7, 2017 at 12:03 pm

Interesting, I would have expected go to be taking some chunks from python (we'll have to watch it over the next few years), but maybe Python is hitting the famous "critical mass." And good on 'em it's a nice language, and much less painful than some others...

**tmh**

September 7, 2017 at 12:27 pm

Suggest me some good sources to learn python. 😊
Thanks.

**Nkansah Rexford**

September 7, 2017 at 8:29 pm

Checkout this: <https://www.python.org/doc/>

**elduderino**

September 7, 2017 at 8:43 pm

Very original link. But a solid choice 😊

**Nkansah Rexford**

September 8, 2017 at 9:09 am

A lesson the Django devs instilled in me. 'RTFM' I was told.

**Roy**

September 8, 2017 at 3:50 pm

How about learning by doing & have fun while you do: codecombat.com
(in addition to going through docs, though)

**Mittens**

September 9, 2017 at 7:43 am

**StianY**

September 7, 2017 at 4:47 pm

Why on earth is java (alone) experiencing a mid summer drop? Is there something with java that just makes it unbearable in the summer heat?

**Brian Keys**

September 7, 2017 at 5:03 pm

School is out, no homework to be done 😊

**StianY**

September 7, 2017 at 5:46 pm

Hah... good point.

**GotBacon**

September 12, 2017 at 9:14 am

Java is pretty much unbearable in any temperature. It's unfortunate that it's taught so widely – it was the de-facto go-to language when the current crop of instructors were learning programming, and so that's why it's being taught now. Pretty much any modern language would be better.

**HerrinSchadenfreude**

September 13, 2017 at 7:56 pm

JAVA is horrific. That's who.

**Bob Foster**

September 7, 2017 at 4:50 pm

The number of StackOverflow questions on Python seems likely to have to do with the quality of Python documentation, which ranges from sketchy to poor. StackOverflow is where a search winds up when answers can't be found elsewhere.

**dan**

September 7, 2017 at 5:30 pm

Python's documentation is no worst than javascript or php. Anyway who reads the docs when you have stackoverflow.

**sth**

September 7, 2017 at 5:35 pm

How would that explain the steady increase of Python questions?

**Leo**

September 7, 2017 at 5:36 pm

Do you read the documentation when you want to solve a programming challenge? Most developers go to stackoverflow to solve their issue quickly and continue working.

I think this is more a cultural thing more than the quality of the documentation. As I see, nowadays the developers prefers fastest answers than exhausting search.

**Another Pt**

September 8, 2017 at 1:25 am

I am a quant not a programmer. So, I asked most dumbest Q in the planet on earliest threads. It's a great community of experts & just attempted to be helpful. Either way, I found it much better than any other. Good lang structure really helps as the basis. Python is for the solvers not the the elite few like others.

**Vince O'Sullivan**

September 9, 2017 at 4:43 am

I hate to say it but that last sentence is about the dumbest comment on Python in the planet.

**Another Pt**

September 9, 2017 at 10:35 am

I don't mind saying it. That comment tells you how little u know about quant analysts and your troubled up bringing. Don't get personal & insulting, unless personally provoked.

**Jason Shin**

September 7, 2017 at 7:14 pm

I think another factor is countries like Australia started to teach their university fundamental programming courses in Python. But I feel like Python is currently in a middle ground between being Node.js and Golang.

**BlueRaja**

September 7, 2017 at 7:40 pm

...What does that even mean? One of those is a technology, the other is a language targeting the complete opposite dev-space as Python

**Nkansah Rexford**

September 7, 2017 at 8:27 pm

"Python is currently in a middle ground between being Node.js and Golang."? And what is that supposed to mean?

**Paula**

September 8, 2017 at 1:04 am

I also would like to see trends in terms of Python 2 versus 3, I feel the latter is finally picking up.

**Loz Cherone**

September 8, 2017 at 1:07 am

More people are learning to hack.. noice.

**Roy**

September 8, 2017 at 3:48 pm

Nope, it's all percentage, so in fact no-one more could be learning to hack today than last year, it might be that less people are asking visiting questions about C#. Who knows, because the author graphed percentages against

**Sergei**

September 8, 2017 at 2:12 am

Where is the Kotlin? it must have experienced an incredible growth this year!

**g00glen00b**

September 8, 2017 at 6:29 am

I was thinking this as well. Even though it isn't a big tag, its growth was quite high:

<https://insights.stackoverflow.com/trends?tags=kotlin>

**André Rebentisch**

September 8, 2017 at 4:46 am

Traffic to Stackoverflow means persons who want answers to questions about Python.

Python is a simple development language but still raises lots of syntax questions. I learned R prior to the year 2000 and came across Python through a friend in 2002. Today their eco systems are completely different, no one says that R is like S anymore, persons formerly known as SPSS professionals became R experts, and Python now means mostly web app development, Django/Flask etc. Syntax and features changed over the years but what remains is that Python is simple but also quite unintuitive. It makes you raise lots of questions.

That could be a super-mario style gamification of learning programming in Python but it is also a result of insufficient official documentation and Python's common oddities.

**Augustin Riedinger**

September 8, 2017 at 9:31 am

I agree one can have doubts on over-interpreting this result.

Eg. Would the curves have similar shape and proportion if we would look at question creations instead of views?

**jam**

September 8, 2017 at 4:27 pm

Python is just about as intuitive as you'd get with a programming language as a beginner. I really disagree with you there. I think the influx of traffic can be explained because python is so intuitive, ie; beginners can pick up python and formulate intermediate questions quickly with python – thus explaining the need of stackoverflow.

**Pavel Goran**

September 9, 2017 at 11:24 pm

I tend to agree with the "simple but unintuitive" statement, although I'm looking from a non-beginner perspective here. For example, using "len(s)" (and not the object attribute/method) to find out the length of a string/list is quite unintuitive because I expect Python to be all-object-oriented; another thing is the need to put the module name in front of a function name (for example, string.split).

**Manoel Vilela**

September 17, 2017 at 5:26 am

Why seems so hard for some people not using brackets to delimit scopes?

**Winston Kotzan**

September 8, 2017 at 12:03 pm

correlation to the language's popularity. As Andre Reberisch suggests in the comment below, it could just mean that Python is harder to learn due to inadequate documentation.

**Ravan Asteris**

September 11, 2017 at 2:14 pm

This.

In spite of several books, the language is nearly gibberish. The type nomenclature sucks, stray whitespace will cause a syntax error (this discourages readability and comments, IMO), and you have to load libraries just to do anything. While this is fine for C (where you have to drag in libraries to write "Hello World"), it is outrageous for a non-compiled scripting language like Python.

Hence, StackOverflow gets more questions as more companies and tools require using it.

**GotBacon**

September 12, 2017 at 9:10 am

Wow. You've obviously never used Python professionally (or even amateur-ally!). It is one of the easiest languages to use, your "stray whitespace" comment is a complete non-issue for anyone that's used it on any bigger program than "Hello World", and it's motto of "batteries included" means that it has a lot more built in functionality than most languages.

**Ravan Asteris**

September 15, 2017 at 9:08 pm

If it wasn't gibberish, I would use it.

I had to use it on an AWS project, and the script I had to expand and maintain was unreadable, uncommented, and was broken every time I tried to comment it and spread it out so it was readable by something other than a 20 year old with no life.

Plus, have you tried to read anaconda? That was my first exposure, and it gave me bad Tcl flashbacks.

Both instances were at paid professional jobs, one in 2002, one in 2014.

Your argument is invalid, because what you state as "obvious" is not true.

**Michael Kennedy**

September 8, 2017 at 12:03 pm

Really nice work guys. We covered this on the Python Bytes podcast this week: <https://pythonbytes.fm/42>

**James**

September 8, 2017 at 1:51 pm

What's the excitement?! Nothing new! All of the you mentioned has already been done in R when I was back in grad school

**szahn**

September 8, 2017 at 2:41 pm

I wonder how much of this growth comes from Python 2 vs Python 3

**mrodent33**

September 8, 2017 at 3:30 pm

This tag trend could indicate many, many things, particularly given the spectrum of experience of SO users. It could indicate so many things that any article wishing to talk about it needs first to consider in some depth what can possibly and realistically be achieved by looking at tag popularity. Down. Vote. (!)

**Grzegorz**

September 8, 2017 at 3:31 pm

Quite interesting job, but is there any correlation to percentage of post that do not include language name? For example I see plenty of times posts about php symfony, doctrine and so on without the "php" tag.

**Roy**

September 8, 2017 at 3:55 pm

Please stop mapping percentages against percentages, as they have little relation to actual growth, being out of 100 as they are.

**David Robinson**

September 11, 2017 at 7:24 pm

All the percentages are out of the same exact denominator, which is the number of visits per month to Stack Overflow. This has been slowly but steadily decreasing over time, with notable drops in each December (associated with the holidays in many Western countries) and to a lesser extent the summer.

Would it really make for a more informative or descriptive plot if each were curve were slating more upwards, with associated seasonal drops? It wouldn't change the relationship between any pair of lines. This graph also makes it clearer which tags are seasonally associated with the school year (Java, C++) which are negatively associated with it (Javascript), and whose traffic is nearly independent of it (Python).

**BASTA!**

September 8, 2017 at 5:47 pm

The only thing StackOverflow question counts indicate directly is that Python is the fastest growing cause of programmers' headaches, which is entirely expected of this monstrosity.

**Lazza**

September 8, 2017 at 7:22 pm

You misunderstood the plots. The "monstrosities" (aka PHP and C++) have a lower question count.

**wpostma**

September 8, 2017 at 6:53 pm

Why not correlate this data with your Jobs/Careers section? Where is there growth in those countries in job postings? There are 1449 jobs on your jobs site. JavaScript brings up 596. Java search brings up 557. Python 428. C# 208. Almost all the jobs on there are in the "Rich Countries" (North America, Europe) and very few are not.

It would be interesting to get some data from your Careers site on the last few years and how many jobs were posted per year for each of these very popular technologies.

**mhmd trbls**

September 9, 2017 at 6:16 am

I will never understand why people choose python over ruby

The increasing interest in data science and the available packages that allow the creation of statistical learning models has a lot to do with the growth of both python and r. Combine that with the straightforward syntax of python and you got yourself a winner.

**Pavel Goran**

September 9, 2017 at 10:57 pm

I don't know about others, but personally I'd steer away from Ruby because of its package management systems (RubyGems and Bundler), which are difficult to understand and use properly, and don't play well with distributions' package managers.

**hipertracker**

September 12, 2017 at 7:32 am

Bundler is very easy to use. It was the inspiration for the others. Now Npm, Yarn, even PHP use similar solutions.

**Håvard Pedersen**

September 13, 2017 at 3:06 am

Unlike pypi?

**Pavel Goran**

September 13, 2017 at 4:57 am

Well, I only used pypi it like one or two times, and I don't remember having problems with it. At least, with pypi, you don't deal with Gemfiles and some weird installation magic to get the application working, you just install the packages that you want and go on.

**Håvard Pedersen**

September 14, 2017 at 3:56 pm

That is literally the exact same thing you do with requirements.txt and pypi.

**Pavel Goran**

September 14, 2017 at 4:39 pm

Not quite. Maybe I was just lucky, but I didn't encounter requirements.txt before. With Ruby, dealing with Gemfiles and Bundler seems unavoidable. And the problem is that Bundler tries to do much more than just installing packages. It manages its own local configuration, mixes application-level configuration with dependency configuration, employs a concept of "groups" to vary installed packages depending on different use cases. Now, there is also "rake" that also has to do with application installation and maintenance. All of this constitutes a complex, very non-trivial ecosystem with varying types of configuration files and whatnot. It may be great and convenient to use if you are a Ruby expert, but if you are just a sysadmin who wants to figure out why Redmine suddenly breaks after Gentoo updates some ruby packages... Well, good luck with that.

**mharris717**

September 14, 2017 at 12:29 pm

Disagree with you here. Ruby's package management solution is one of its best features, and (just IMO) superior to Python's. Generally, of course, which language you prefer is mostly a preference, and they're both great languages.

**emrah**

September 10, 2017 at 11:21 am

**Ageofwant**

September 10, 2017 at 11:23 am

Python is dominant in science and big data, also older, so more established. Python also has a superior syntax and data model. Ruby is great, but Python is better.

**hipertracker**

September 11, 2017 at 10:59 am

Python is much less expressive language than Ruby. Python's syntax is also uglier. But... Python has several very good libs like SciPy and is much simpler language to start with.

**Manoel Vilela**

September 17, 2017 at 5:22 am

The `end` keyword.

**Eric Vergnaud**

September 9, 2017 at 7:01 am

Well there is a major bias in this paper, where the writer makes the assumption that the number of questions reflects the growth of usage.

While there is certainly some correlation, it may very well be that growth of questions relates to many other factors:

- growth of problems. It is easy to imagine that if you don't have problems, you won't ask questions. It may be that with Python apps gradually growing in size and scope, more problems appear than with other languages
- questions already answered. You can find answers to many questions in say Java, hence new users won't ask new questions

I am not expressing an opinion on Python here, simply an opinion on the paper. Growth of questions does not translate to growth of usage (unless proven otherwise)

**maxnoe**

September 9, 2017 at 8:23 am

It's question views, not questions asked. Existing questions thus count as well as new ones.

**Voros Rukmini Xanathor**

September 9, 2017 at 7:32 pm

I will make FORTRAN pop again

**doktor**

November 3, 2017 at 1:54 am

Sure, but please at least spell it "Fortran" as it has been spelled for over a quarter century. 🙄

**Abishek Muthian**

September 9, 2017 at 11:12 pm

It would be interesting to collect data to see how many continued using python from prototype to production. Python would definitely help to build MVP, when in production where actual concurrency matters; very few like Instagram have built robust python based architectures.

Btw, achieving decent concurrency in python is quite straightforward. Understanding of Greenlets, Gevent, uvloop Cython etc. could prevent headache in future.

Seems the researcher need to understand how Stack Overflow is used before making such a misleading statement. A higher score on Stack Overflow Trends would indicate the inadequacies of the language. More visits indicate a level of frustration, a lack of documentation, and not the languages popularity.

Traffic from high-income countries (US/UK) is also misleading, without understanding the root cause. Python started out as an ASCII only language, making it useless in non-ASCII environments. In lower-income (non-English speaking) countries, they do not choose Python because many outputs default to the ASCII character set.

Using Python takes a constant effort to use a character set that's not ASCII. Keeping the desired character set is a constant fight, one I have personally faced time and time again. Adding a lot of extra code that is not necessary in other programming languages.

**a**

September 10, 2017 at 5:37 am

python3 is better with unicode default

**Ageofwant**

September 10, 2017 at 11:18 am

Sorry no, this is wrong on all accounts. Even ancient Python supported unicode out of the box. Python 3, which is 11 years old, uses utf-8 source encoding and unicode strings by default.

**Daniel Sabinasz**

September 10, 2017 at 10:09 am

Maybe it's not the number of Python users that increased, but the confusion?

**Khalil Al Hooti**

September 10, 2017 at 5:29 pm

then Java is the most confusing language of all times.

**Manoel Vilela**

September 17, 2017 at 5:29 am

But... this is a quite true :v

**Manoel Vilela**

September 16, 2017 at 2:44 pm

Yes, probably. Actually I believe is just people trying learn how to use third-libraries of WebScrapping and Machine Learning works (pandas, sklearn, tensorflow, theano and etc), which in essence are complex things and not exactly because of Python, but their itself embedded technologies.

**Daniel Sabinasz**

September 16, 2017 at 5:03 pm

It's probably a combination of both. The recent machine learning / deep learning hype has brought into existence many new ML Python libraries that need to be learned by users, even if they were Python users before, so it increases the number of questions they ask. Additionally, the fact that most popular machine learning libraries are written in Python may also cause more people to use Python.

**Gordon Messmer**

September 10, 2017 at 12:17 pm

**Foo**

September 11, 2017 at 6:01 am

I don't think that's right. I just checked in a dictionary and "Twofold" means doubling and "threefold" means tripling. I'm pretty sure 2.5-fold wouldn't mean 5.6 times much.

**Gordon Messmer**

September 11, 2017 at 4:24 pm

That dictionary may also tell you that the meaning of "literally" is "not literally." They reflect common use, which in some cases includes common inaccurate word use.

https://en.wikipedia.org/wiki/Fold_change

**Tal Friedman**

September 11, 2017 at 8:14 pm

That's a usage that I haven't seen even in technical stats literature – seems pretty esoteric. But either way by that definition 2.5-fold is exactly 3.5 times as much as the original value (it's just the difference between new and old in terms of the old). I'm not sure where you got 5.6 times, maybe you interpreted it as being something exponential?

**Khalil Al Hooti**

September 10, 2017 at 5:20 pm

I started My PhD a year and half ago. Programming was something new to me. In the department I am in most postdoc use python. I started learning it. Believe me. this language is amazing.

**Sid1138**

September 10, 2017 at 6:06 pm

An interesting analysis, but one problem with statistical analysis is assigning a causation to a correlation. So, the correlation is one of number of views. The causation assumption is because of the number of users. However, there are other potential causation theories, such as lack of documentation, inexperience of users/viewers, source of viewership (e.g. students versus professionals), etc.

I always find it disturbing when people jump to causation theories based on a set of statistics when the full set of statistics are not identified (and often not even available). This can lead to some erroneous conclusion that people then base actions on incorrectly. One of my favorite examples is car insurance in the USA. In the '80s, red-colored cars were in more accidents than any other color car. The conclusion reported in the US news media was that a red-colored car was more dangerous than other color. Legislation was proposed to raise the insurance costs of those who owned red cars (luckily, this did not go through).

What question was not addressed by the statistics? How many red cars were out there. It turned out that there were way more red cars than any other color. When you adjusted for number of cars, it actually turned out that the red cars were in fewer accidents per car than any other color. So, instead of being the most dangerous, they were the safest (at least when counting number of accidents, there were no statistics on fatalities per color that I could find).

Pushing correlation as causation is a favorite past time of statisticians and those that report statistics. I suggest that while reviewing the numbers is very interesting, take the causation statements with a grain of salt.

**Paul Clinch**

September 11, 2017 at 6:11 am

The red cars point is just terrible use of statistics, its not related to the correlation as causation problem.

Paul, you are correct that the red car was a terrible use of statistics. However, the problem is the same. Someone sees a correlation in the statistical set they are reviewing and assign a causation (implied or stated) to that correlation. In this case the correlation was number of question views over time and the assumed causation is a significant increase in python programmers and language use. For example, the numbers don't state "unique user views".

Anyway, I do like Stack Overflow's statistics, I just don't base my career decisions on just that data set.

**J.F. Sebastian**

September 16, 2017 at 2:10 am

"Conclusion: do questions really represent the health of a technology?" <https://stackoverflow.blog/2017/08/01/flash-dead-technologies-might-next/>

**Abishek Muthian**

September 10, 2017 at 10:19 pm

It would be interesting to collect data to see how many continued using python from prototype to production. Python would definitely help to build MVP, when in production where actual concurrency matters; very few like Instagram have built robust python based architectures.

Btw, achieving decent concurrency in python is quite straightforward. Understanding of Greenlets, Gevent, uvloop Cython etc. could prevent headache in future.

**Viet Vu**

September 10, 2017 at 10:28 pm

With the raise of machine learning, AI, data science.. it's no surprise to me. There was none Python programmer in my company before except for me. Now for our new R&D, BI, the main/only language we are using is Python.

**Chris Matichuk**

September 12, 2017 at 11:00 am

If developer time is more valuable than compute costs, than it's about the effort to code the solution – and for ML, Python appears to be the preferred choice.

**waynewerner**

September 13, 2017 at 11:38 am

I'd love to see what sort of tags within Python are seeing the most visits/growth. I'm betting that there's a lot of pandas and other scientific Python driving these visits.

**Asanka Jayasundara**

September 15, 2017 at 10:48 pm

most probably sklearn,pandas and matplotlib

**sd**

September 23, 2017 at 10:35 am

ruby is better

**#DraftHillary2020**

September 27, 2017 at 4:10 pm

thing.

I use C# 40 hours a week at my day job, and Python for a data science class (~10 hrs week). I can safely say I visit StackOverflow a lot more often with Python questions.



Michael Kennedy

September 27, 2017 at 4:46 pm

You should use PyCharm if you prefer the C# experience. It's like Visual Studio + Resharper only better.



#DraftHillary2020

September 30, 2017 at 8:19 am

Nice!



Another Pt

October 28, 2017 at 7:14 pm

Just started on Julia. Wow! this is an excellent language. I now know it better after a week than Scala after months of investment. If anything can dethrone R, Python, Scala, Matlab, etc..., then Julia may be able to. Took the nearly best features. I do miss the obj orientation, but really great. This is coming from 2 decades on Python by a member from "Liar's Poker" quant.


Recent Tweets



Stack Overflow
[@StackOverflow](#)

Ever ask someone a question in Slack or email and get information overload? Or answers that lead to more questions?...
<https://t.co/two0Hvgdzb>

7:23 PM - 8 May 2018

A very interesting and introspective thread  <https://t.co/py8iyoNxAL>

3:32 PM - 3 May 2018

Have questions about Stack Overflow for Teams? Ask @spolsky in our @ProductHunt discussion: <https://t.co/AYLDD7noK1>

2:36 PM - 3 May 2018

Developer Jobs

[See all job openings now!](#)

Stack Overflow Talent

Source, attract, and recruit technical talent on the platform developers trust.

[Learn More](#)

STACK OVERFLOW

[Questions](#)
[Jobs](#)
[Developer Jobs Directory](#)
[Salary Calculator](#)

PRODUCTS

[Teams](#)
[Talent](#)
[Ads](#)
[Enterprise](#)

COMPANY

[About](#)
[Press](#)
[Work Here](#)
[Legal](#)
[Privacy Policy](#)
[Contact Us](#)

STACK EXCHANGE NETWORK

[Technology](#) >
[Life / Arts](#) >
[Culture / Recreation](#) >
[Science](#) >
[Other](#) >

[Blog](#) [Facebook](#) [Twitter](#) [LinkedIn](#)

site design / logo © 2018 Stack Exchange Inc; user
contributions licensed under cc by-sa 3.0 with attribution
required. rev 2018.5.8.30324