

[insights](#)

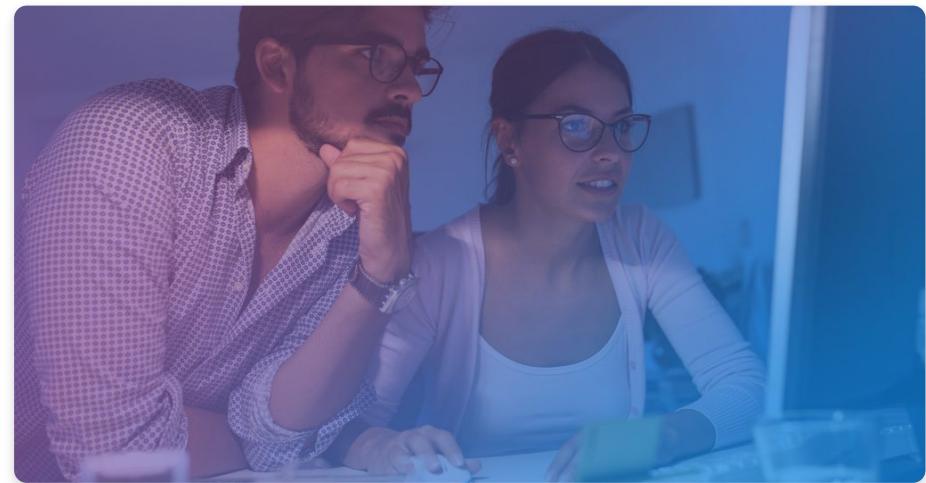
SEPTEMBER 6, 2017

The Incredible Growth of Python

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...

**David Robinson**

Data Scientist (former)



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).

The rent is too damn high.

Search jobs by salary on Stack Overflow Jobs



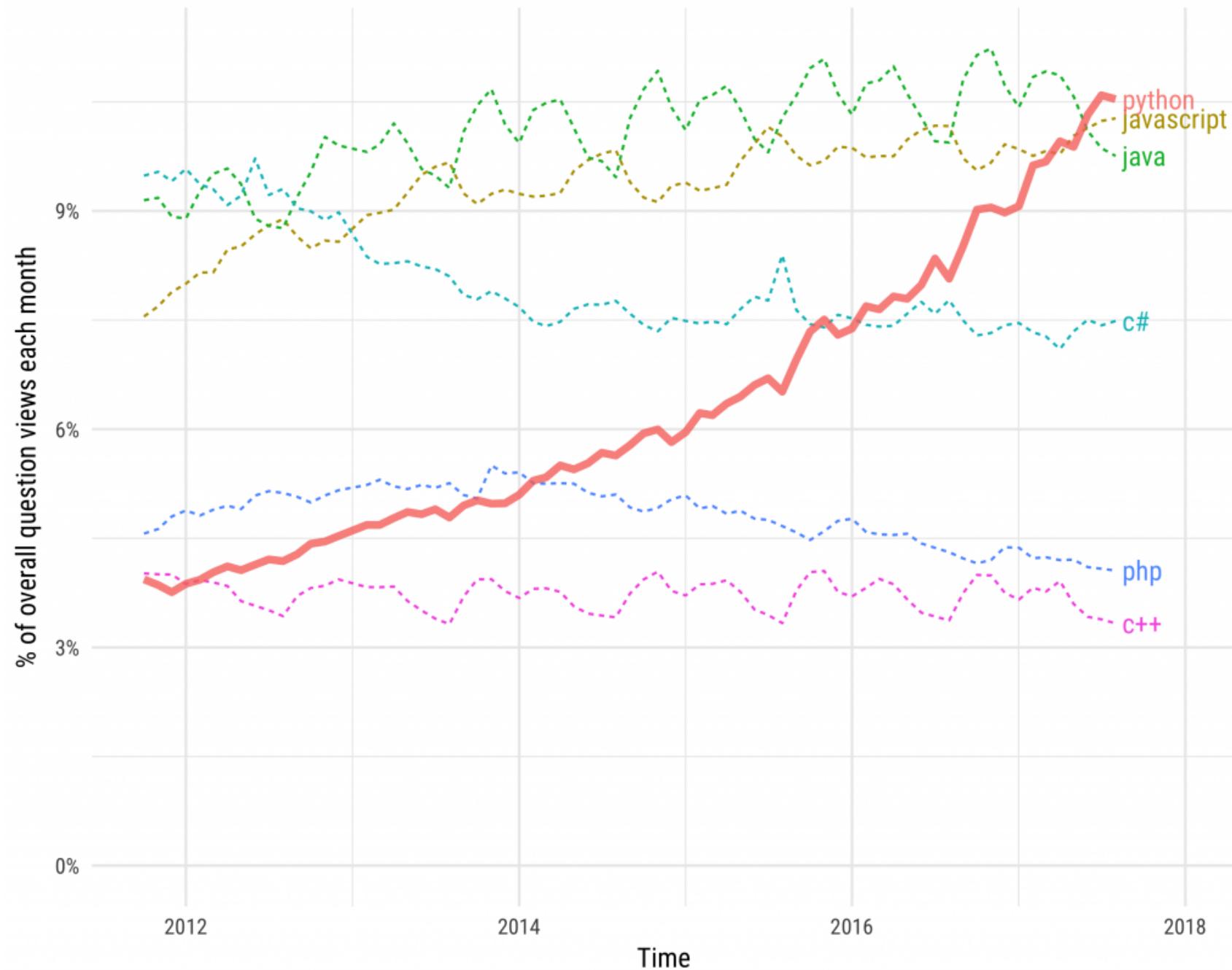
Python's growth in high-income countries

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.

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries

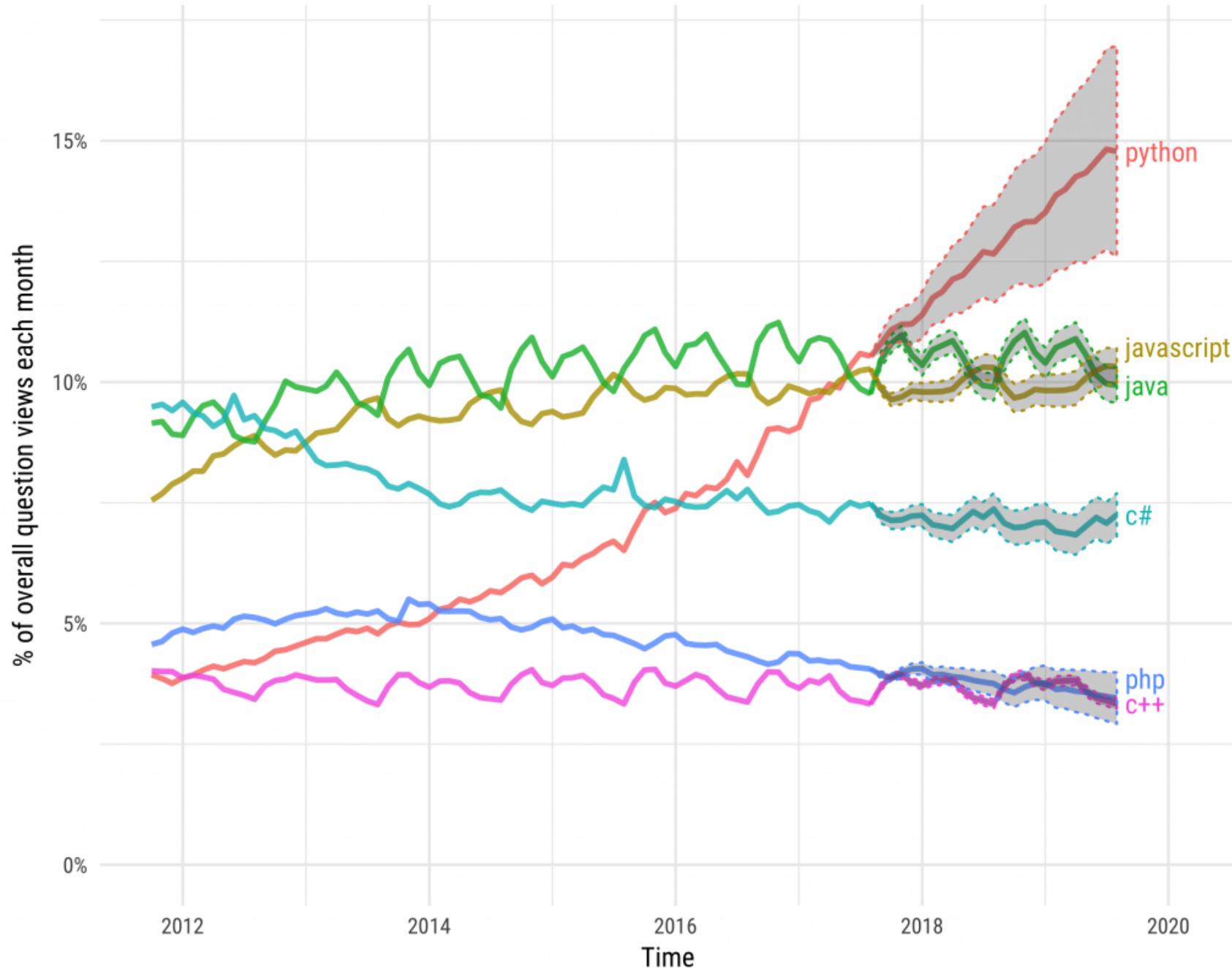


June 2017 was the first month that **Python was the most visited tag on Stack Overflow within high-income nations**. This included being the most visited tag within the US and the UK, and in the top 2 in almost all other high income nations (next to either Java or JavaScript). This is especially impressive 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.

Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.



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?

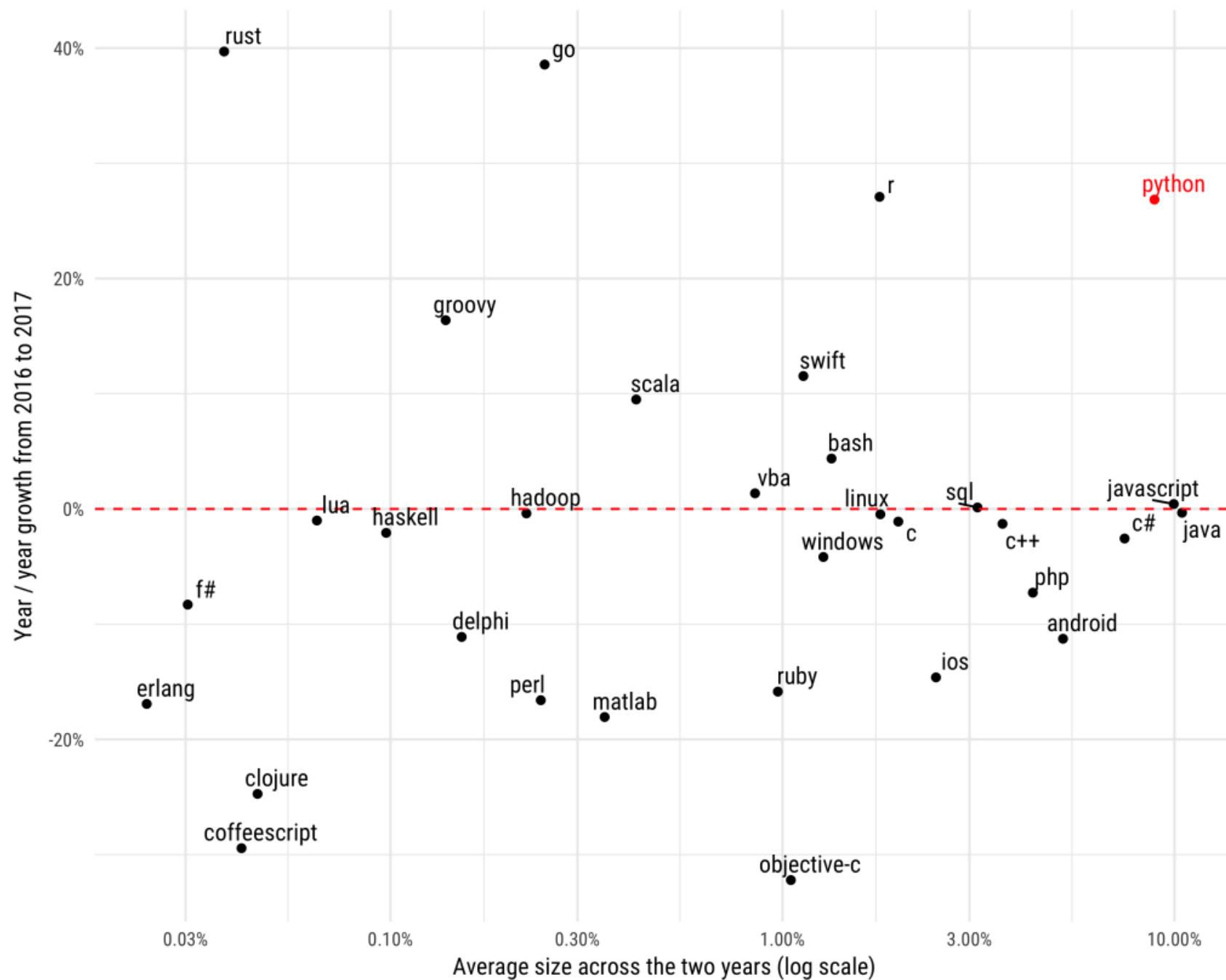
The above was looking only at the six most-visited programming languages. Among other notable technologies, which are currently growing the fastest in high-income countries?

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](#).

Year over year growth in traffic to programming languages/platforms

Comparing question views in January-August of 2016 and 2017, in World Bank high-income countries.
TypeScript had a growth rate of 142% and an average size of .36%; and was omitted.

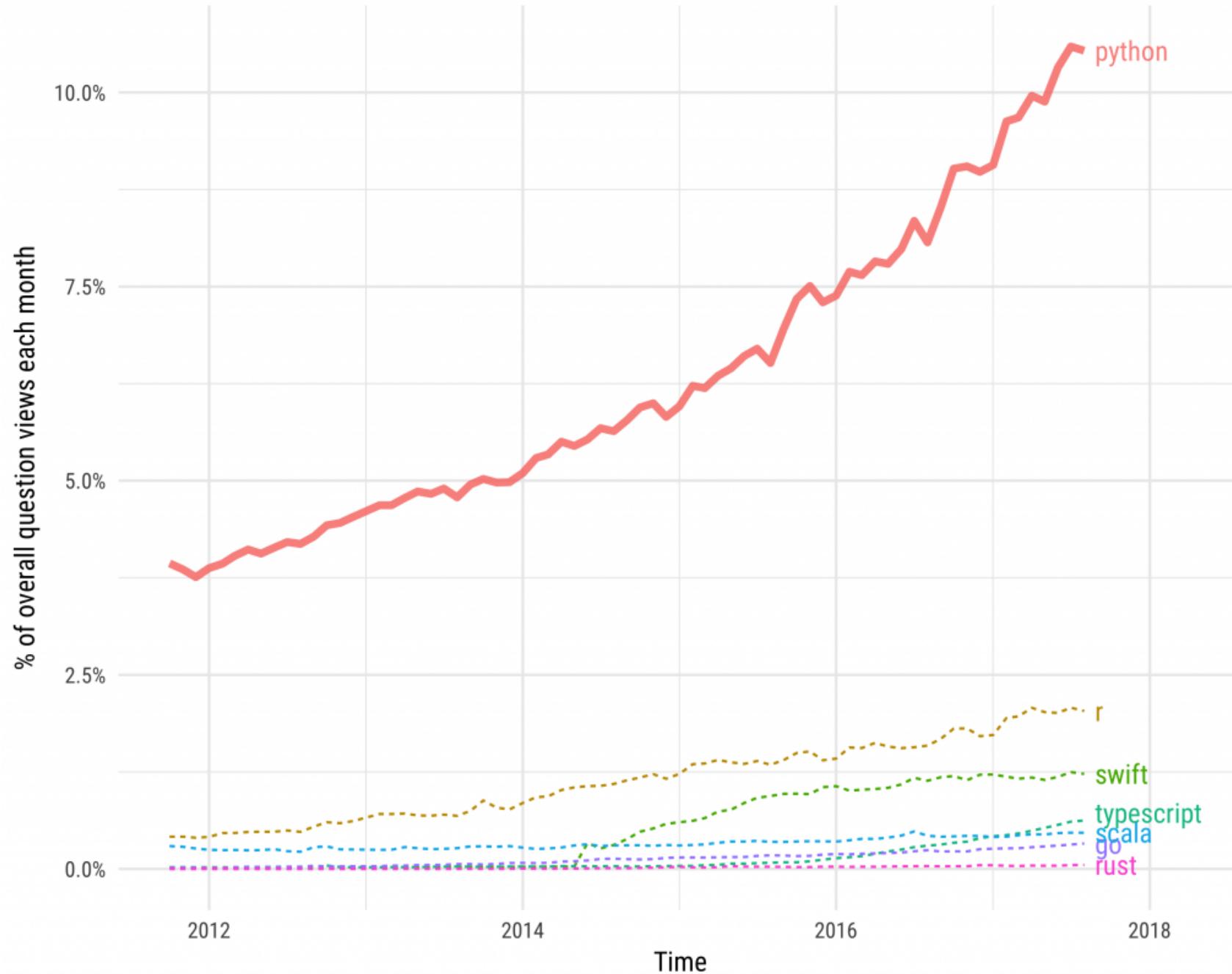


With a 27% year-over year-growth rate, Python stands alone as a tag that is **both large and growing rapidly**; the next-largest tag that shows similar growth is R. We see that traffic to most other large tags has stayed pretty steady within high-income countries, with visits to Android, iOS, and PHP 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?

Python compared to smaller, growing technologies

Based on question traffic in World Bank high-income countries



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).

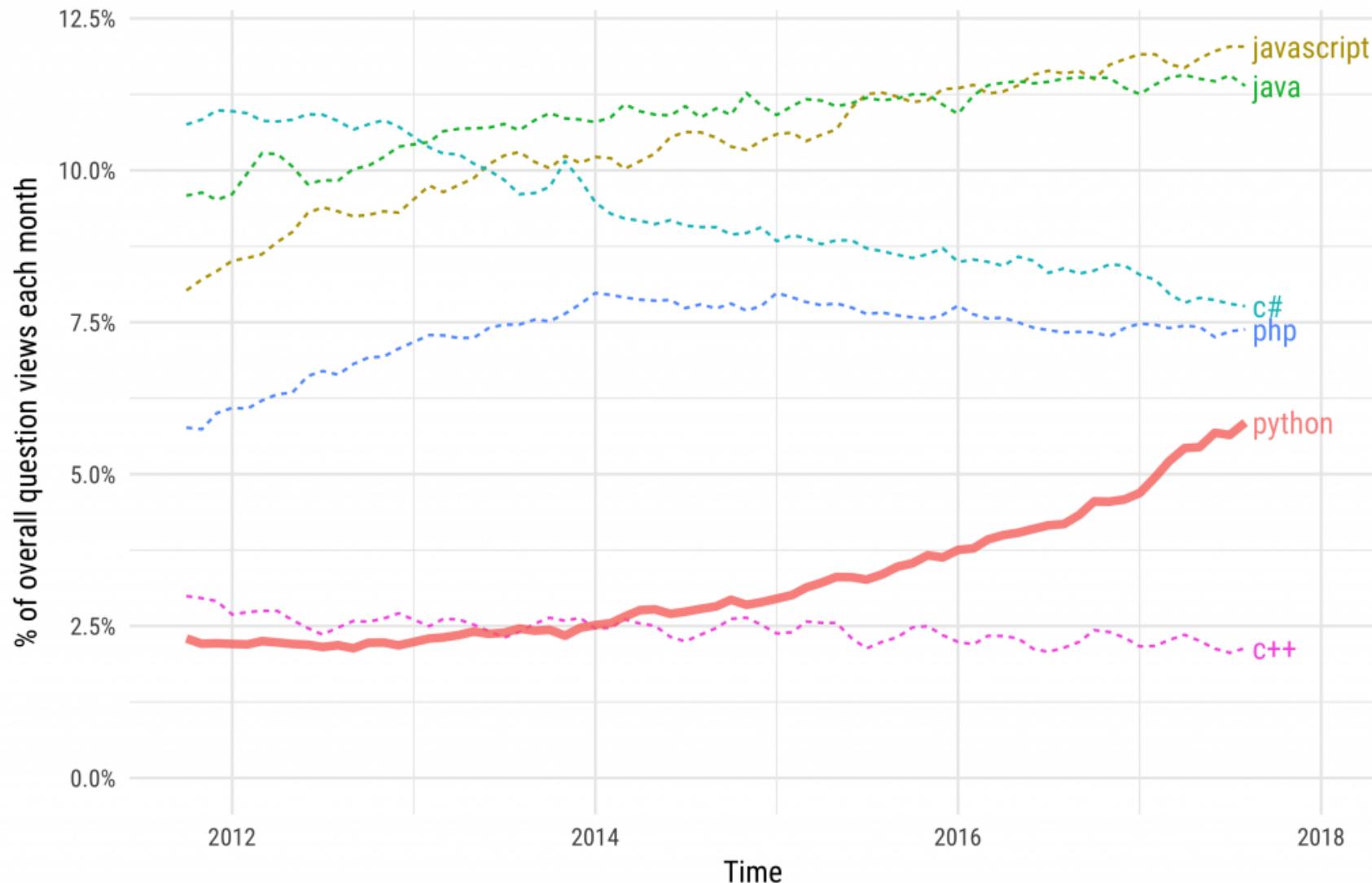
Rest of the world

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.

Growth of major programming languages in non-high-income countries

Based on Stack Overflow question views in countries not classified as high-income by the World Bank.



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).

In the meantime, if you work in Python and are looking to take the next step in your career, here are [some companies hiring Python developers right now on Stack Overflow Jobs](#).

Our related podcast with Dries Buytaert

The Stack Overflow Podcast | EP244

Dropping knowledge with Drupal's creator, Dries

00:00 16:08

1X

SHARE SUBSCRIBE

A screenshot of a podcast player interface. At the top, it says "The Stack Overflow Podcast | EP244". Below that is the title "Dropping knowledge with Drupal's creator, Dries". In the center is a play button with a white triangle pointing right. To its left is a yellow "00:00" button. To its right is a grey waveform representing the audio. On the far right of the waveform is a "16:08" button. Below the play button is a "1X" button. At the bottom right of the player are "SHARE" and "SUBSCRIBE" buttons.



The Stack Overflow Podcast is a weekly conversation about working in software development, learning to code, and the art and culture of computer programming.

The Stack Overflow Podcast | EP251

How to interpret the compiler

00:00

1X

A screenshot of a dark-themed podcast player interface. At the top, it says "The Stack Overflow Podcast | EP251". Below that is the title "How to interpret the compiler". In the center is a large white play button with a black triangle pointing right. To its left is a yellow "00:00" button. To its right is a grey waveform representing the audio. Below the play button is a "1X" button.

Related



community MAY 27, 2020

The 2020 Developer Survey results are here!

We're excited to share the results of our 10th annual developer survey! 65,000 developers shared their thoughts on the state of software today.



talent FEBRUARY 24, 2020

How Your Tech Stack Can Attract Top Developers

For developers, the single biggest factor in evaluating a job offer is the technology stack they'll be using. The tools and technologies selected as part of a company's core tech stack will heavily impact that company's ability to develop and deploy code quickly and to scale seamlessly. Know what to highlight to candidates about your tech stack.

 **Ben Popper**
Director of Content

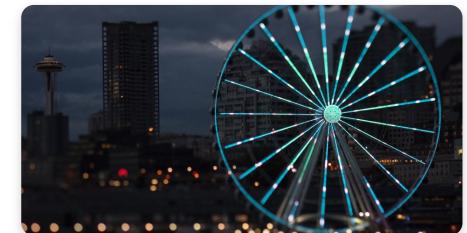
 **Kathryn Hawkins**



company JANUARY 21, 2020

Scripting the Future of Stack Overflow

A note from our new CEO, Prashanth Chandrasekar, reflecting on his first 90 days and laying out his vision for 2020.



code-for-a-living JUNE 15, 2020

Talking TypeScript with the engineer who leads the team

We wanted to find out what about TypeScript makes it so dang lovable, so we reached out to Ryan Cavanaugh, the principal engineering lead for the TypeScript language at Microsoft. He was generous enough to sit down and answer our questions.

 **Prashanth Chandrasekar**
CEO

 **Ryan Donovan**

223 Comments



thecity2

6 Sep 17 at 1:44

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.

Reply



Herr Derb

6 Sep 17 at 4:30

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.

Reply



Max Malysh

6 Sep 17 at 5:11

The first version of Google was written in Python.

Reply



discucus

12 Sep 17 at 4:44

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.

Reply



jgmitzen

6 Sep 17 at 6:56

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>

Reply



Alex Punnen

6 Sep 17 at 11:25

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.

Reply



seagreg

7 Sep 17 at 2:58

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

12 Sep 17 at 2:03

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

6 Sep 17 at 11:23

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.

Reply



Giorgi Moniava

7 Sep 17 at 6:06

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

Reply



new year

24 Sep 17 at 3:28

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.

Reply



Vince O'Sullivan

24 Sep 17 at 5:13

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/>

Reply



James Colin

16 Apr 19 at 5:21

Apr 2019 Apr 2018 Change Programming Language Ratings Change
1 1 Java 15.035% -0.74%
2 2 C 14.076% +0.49%
3 3 C++ 8.838% +1.62%
4 4 Python 8.166% +2.36%



Dan

26 Jul 18 at 6:17

As far as industry goes, Ignition is Python (not CPython but Jython) so you get a lot of controls engineers using Python. You also get a lot of data scientists using it. When I worked for a logistics company, the stack was .Net, but all our data

scientists used Python for predictive analytics, geographic measurements and processing data feeds. While a lot of small businesses use WordPress, you see some small and medium going the Python route via Django.

Reply



camden_kid

6 Sep 17 at 2:55

Fascinating. I've always been interested in the rise and fall in popularity of programming languages.

Reply



Tadeu Manoel

6 Sep 17 at 3:03

Where is Julia? 😊

Reply



David Robinson

6 Sep 17 at 3:26

<https://uploads.disquscdn.com/images/c98fb169ba110c6bd1c81d0865ad13eb420e138ba753343dc30e06c3a6550aa.png>

Reply



aNoobNoob

6 Sep 17 at 3:41

What about Elixir 😊

Reply

**Willy Karam**

6 Sep 17 at 10:57

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

Reply

**David Robinson**

7 Sep 17 at 4:39

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/3593321bf514836d91cc43842b035a2f03ea2bc09373270076ccc34602>

**Willy Karam**

7 Sep 17 at 5:06

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 the cycle. Given that it's later release than Rust/Go, it'll be interested 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**

6 Sep 17 at 4:26

Yiiiikes.. it's still... not even...

Reply



Willy Karam

6 Sep 17 at 10:57

Where is Elixir? 😊

Reply



NewWorld

7 Sep 17 at 7:53

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

Reply



Bill Sommers

7 Sep 17 at 4:35

nice

Reply



Jerry Wasinger

6 Sep 17 at 3:19

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.

Reply



Izkata

6 Sep 17 at 4:02

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](#)

Reply



Kenneth J Hughes

6 Sep 17 at 4:54

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

Reply



Alec

6 Sep 17 at 4:56

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?

Reply



Mike Croft

6 Sep 17 at 5:04

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.

Reply



Bryan Oakley

6 Sep 17 at 5:59

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.

Reply

**Manthan Thakar**

6 Sep 17 at 5:05

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.

[Reply](#)**Mark Hancock**

6 Sep 17 at 10:26

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

[Reply](#)**Ravan Asteris**

12 Sep 17 at 1:59

That's because space delimited languages discourage commenting.

[Reply](#)**js**

7 Sep 17 at 9:39

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

[Reply](#)

7 Sep 17 at 10:22



sth

1 Sep 17 at 12:53

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.

Reply



Alex P

6 Sep 17 at 5:08

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.

Reply



microamp

6 Sep 17 at 6:42

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.

Reply



merito

6 Sep 17 at 10:29

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

Reply

**wonho ha**

6 Sep 17 at 6:59

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

[Reply](#)**asd**

7 Sep 17 at 9:18

like Nsync vs. backstreet years ago

[Reply](#)**Nick Feller**

6 Sep 17 at 7:33

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

[Reply](#)**Dr Porostertartamunduskusz**

8 Sep 17 at 7:31

It is also interesting that the JS chart has its peaks when Java has its lows. So those students start to go wild with JS during their break times?

[Reply](#)**Sillas Gonzaga**

6 Sep 17 at 9:25

Is the code for this analysis available?

Reply



Naranmandakh Tsogoo

7 Sep 17 at 1:09

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

Reply



Thomas Gütter

7 Sep 17 at 1:29

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.

Reply



Ryan Graham

7 Sep 17 at 1:59

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.

Reply



Cpt Vineum

7 Sep 17 at 4:14

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

Reply



smitty

7 Sep 17 at 11:39

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

Reply



Hridayesh Sharma

7 Sep 17 at 4:51

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.

Reply



FDominicus

7 Sep 17 at 4:58

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

Reply



ALT

7 Sep 17 at 5:10

That typescript projection is way off. See angular.

Reply



David Robinson

7 Sep 17 at 8:04

There is no projection for TypeScript.

Reply

**Vivek Jain**

7 Sep 17 at 5:16

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?

[Reply](#)**js**

7 Sep 17 at 9:31

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.

[Reply](#)**amz3**

7 Sep 17 at 5:39

Looking forward the next post!

[Reply](#)**neelkadia**

7 Sep 17 at 5:46

AI and ML are coming bro!

[Reply](#)**Giorgi Moniava**

7 Sep 17 at 6:03

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

Reply



aCppDev

7 Sep 17 at 6:34

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++.

Reply



Iron Sean

7 Sep 17 at 3:13

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

Reply



Toby

7 Sep 17 at 6:45

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 summer break periods. With the acuteness of Python’s trend though it’s hard to tell if the same trend is appearing there yet.

Reply



Philip Bayer

7 Sep 17 at 10:04

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

Reply



BlueRaja

7 Sep 17 at 7:43

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.

Reply



Ettore Rizza

7 Sep 17 at 7:55

"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.

Reply



Lazza

8 Sep 17 at 7:28

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).

Reply



yanggeorge

7 Sep 17 at 9:47

R is bad.

Reply



215756347@qq.com

7 Sep 17 at 9:16

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

Reply



Another Pt

7 Sep 17 at 9:57

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).

Reply



smitty

7 Sep 17 at 11:38

Guido is a BeauDiFuL man.

Reply

**Another Pt**

8 Sep 17 at 1:19

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.

[Reply](#)**James Howe**

7 Sep 17 at 11:41

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

[Reply](#)**Richard Rast**

7 Sep 17 at 12:09

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

[Reply](#)**Steve**

7 Sep 17 at 10:58

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

[Reply](#)**sth**

7 Sep 17 at 12:02

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

Reply



Paula

8 Sep 17 at 1:03

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

Reply



alanjds

7 Sep 17 at 12:50

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

Reply



P.B. Lecavalier

7 Sep 17 at 1:04

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 do 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.

Reply



Ravan Asteris

12 Sep 17 at 1:53

Hahahahaha.

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

Reply



alanjds

13 Sep 17 at 1:50

Yeah. Sure. Sphinx usually indexes it badly. Lets get the flames started :/

Reply



John Rosewicz

7 Sep 17 at 1:41

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.

Reply



Saelyth

7 Sep 17 at 2:56

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.

Reply



Roger Delyo Pack

7 Sep 17 at 12:03

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...

Reply



tmh

7 Sep 17 at 12:27

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

Reply



Nkansah Rexford

7 Sep 17 at 8:29

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

Reply



elduderino

7 Sep 17 at 8:43

Very original link. But a solid choice 😊

Reply



Nkansah Rexford

8 Sep 17 at 9:09

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

Reply



Roy

8 Sep 17 at 3:50

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

Reply



Mittens

9 Sep 17 at 7:43

SoloLearn – if you already know how to program. They give you a nice run-through of the major features.

Reply



StianY

7 Sep 17 at 4:47

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?

Reply



Brian Keys

7 Sep 17 at 5:03

School is out, no homework to be done 😊

Reply



StianY

7 Sep 17 at 5:46

Hah... good point.

Reply



GotBacon

12 Sep 17 at 9:14

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.

Reply



HerrinSchadenfreude

13 Sep 17 at 7:56

JAVA is horrific. That's who.

Reply



Bob Foster

7 Sep 17 at 4:50

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.

Reply



dan

7 Sep 17 at 5:30

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

Reply



sth

7 Sep 17 at 5:35

How would that explain the steady increase of Python questions?

Reply



Leo

7 Sep 17 at 5:36

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.

Reply



Etra

7 Sep 17 at 10:49

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.

Reply



Another Pt

8 Sep 17 at 1:25

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.

Reply



Vince O'Sullivan

9 Sep 17 at 4:43

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

Reply



Another Pt

9 Sep 17 at 10:35

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.

Reply



Jason Shin

7 Sep 17 at 7:14

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.

Reply



BlueRaja

7 Sep 17 at 7:40

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

Reply



Nkansah Rexford

7 Sep 17 at 8:27

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

Reply

**Paula**

8 Sep 17 at 1:04

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

[Reply](#)**Loz Cherone**

8 Sep 17 at 1:07

More people are learning to hack.. noice.

[Reply](#)**Roy**

8 Sep 17 at 3:48

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 percentages within a closed system!

[Reply](#)**Sergei**

8 Sep 17 at 2:12

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

[Reply](#)**g00glen00b**

8 Sep 17 at 6:29

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>

Reply



André Rebentisch

8 Sep 17 at 4:46

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.

Reply



Augustin Riedinger

8 Sep 17 at 9:31

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?

Reply



jam

8 Sep 17 at 4:27

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.

Reply



Pavel Goran

9 Sep 17 at 11:24

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`).

Reply



Manoel Vilela

17 Sep 17 at 5:26

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

Reply



Winston Kotzan

8 Sep 17 at 12:03

This clickbait article sucked. I expected it to go into detail about the business drivers of why organizations might be adopting Python over other languages. Instead it just graphed the number of tech support questions, suggesting a correlation to the language’s popularity. As André Rebentisch suggests in the comment below, it could just mean that Python is harder to learn due to inadequate documentation.

Reply



Ravan Asteris

11 Sep 17 at 2:14

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.

Reply



GotBacon

12 Sep 17 at 9:10

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 its motto of “batteries included” means that it has a lot more built in functionality than most languages.

Reply



Ravan Asteris

15 Sep 17 at 9:08

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.

Reply

**Michael Kennedy**

8 Sep 17 at 12:03

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

[Reply](#)**James**

8 Sep 17 at 1:51

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

[Reply](#)**szahn**

8 Sep 17 at 2:41

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

[Reply](#)**mrodent33**

8 Sep 17 at 3:30

This article is incorrectly and sensationalized titled: it's not worthy of an "official" SO article. It should make abundantly plain that it is about the popularity of tags on SO, not something doing in-depth research into which languages are being used.
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. (!)

[Reply](#)**Grzegorz**

8 Sep 17 at 3:31

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.

Reply



Roy

8 Sep 17 at 3:55

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

Reply



David Robinson

11 Sep 17 at 7:24

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).

Reply



BASTA!

8 Sep 17 at 5:47

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.

Reply



Lazza

8 Sep 17 at 7:22

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

Reply



wpostma

8 Sep 17 at 6:53

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.

Reply



mhmd trbls

9 Sep 17 at 6:16

I will never understand why people choose python over ruby

Reply



fasdfs

9 Sep 17 at 3:53

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.

Reply



Pavel Goran

9 Sep 17 at 10:57

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.

Reply



hipertracker

12 Sep 17 at 7:32

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

Reply



Håvard Pedersen

13 Sep 17 at 3:06

Unlike pypi?

Reply



Pavel Goran

13 Sep 17 at 4:57

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.

Reply



Håvard Pedersen

14 Sep 17 at 3:56

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

**Pavel Goran**

14 Sep 17 at 4:39

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**

14 Sep 17 at 12:29

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.

[Reply](#)**emrah**

10 Sep 17 at 11:21

Ruby and Python are not at the same level. Maybe Ruby over php.

[Reply](#)**Ageofwant**

10 Sep 17 at 11:23

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.

Reply



hipertracker

11 Sep 17 at 10:59

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.

Reply



Manoel Vilela

17 Sep 17 at 5:22

The `end` keyword.

Reply



Eric Vergnaud

9 Sep 17 at 7:01

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)

Reply



maxnoe

9 Sep 17 at 8:23

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

Reply



Voros Rukmini Xanathor

9 Sep 17 at 7:32

I will make FORTRAN pop again

Reply



doktor

3 Nov 17 at 1:54

Sure, but please at least spell it “Fortran” as it has been spelled for over a quarter century. 😊

Reply



Abishek Muthian

9 Sep 17 at 11:12

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.

Reply



Brian LaVallee

10 Sep 17 at 12:54

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.

Reply



a

10 Sep 17 at 5:37

python3 is better with unicode default

Reply



Ageofwant

10 Sep 17 at 11:18

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.

Reply



Daniel Sabinasz

10 Sep 17 at 10:09

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

Reply



Khalil Al Hooti

10 Sep 17 at 5:29

then Java is the most confusing language of all times.

Reply



Manoel Vilela

17 Sep 17 at 5:29

But... this is a quite true :v

Reply



Manoel Vilela

16 Sep 17 at 2:44

Yes, probably. Actually I believe is just people trying learn how to use third-libraries of WebScraping 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.

Reply



Daniel Sabinasz

16 Sep 17 at 5:03

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.

Reply

**Gordon Messmer**

10 Sep 17 at 12:17

Nitpick: Fold growth represents doubling. 2.5-fold is roughly 5.6 times as much as the original value. Python is seeing about 2.5 times as much traffic, not 2.5-fold as much traffic.

[Reply](#)**Foo**

11 Sep 17 at 6:01

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.

[Reply](#)**Gordon Messmer**

11 Sep 17 at 4:24

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

[Reply](#)**Tal Friedman**

11 Sep 17 at 8:14

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?

[Reply](#)

**Khalil Al Hooti**

10 Sep 17 at 5:20

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.

[Reply](#)**Sid1138**

10 Sep 17 at 6:06

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.

[Reply](#)**Paul Clinch**

11 Sep 17 at 6:11

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

[Reply](#)

**Sid1138**

11 Sep 17 at 2:08

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.

[Reply](#)**J.F. Sebastian**

16 Sep 17 at 2:10

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

[Reply](#)**Abishek Muthian**

10 Sep 17 at 10:19

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.

[Reply](#)**Viet Vu**

10 Sep 17 at 10:28

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.

Reply



Chris Matichuk

12 Sep 17 at 11:00

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.

Reply



waynewerner

13 Sep 17 at 11:38

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.

Reply



Asanka Jayasundara

15 Sep 17 at 10:48

most probably sklearn,pandas and matplotlib

Reply



sd

23 Sep 17 at 10:35

ruby is better

Reply



#DraftHillary2020

27 Sep 17 at 4:10

I think this methodology works when comparing a single language's growth, but not when comparing languages with each other. For example when I use C#, probably 80% of my questions are answered by intellisense. To the extent that I've used Python, it has typically been inside a Jupyter notebook, where you have to look up everything little 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.

Reply



Michael Kennedy

27 Sep 17 at 4:46

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

Reply



#DraftHillary2020

30 Sep 17 at 8:19

Nice!

Reply



Another Pt

28 Oct 17 at 7:14

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.

Reply

**MW**

10 Feb 19 at 3:51

no doubt Julia has pros, but Pascal-like syntax killed my interest in this language...

[Reply](#)**Mark**

13 Jul 18 at 2:00

Why do you think there is such a strong seasonality for some of the languages like java, javascript, and c++ in the high-income countries?

[Reply](#)**Saranya Varma**

9 Mar 19 at 9:49

Ofcourse, wealthy countries are moving towards much of the Artificial Intelligence and Cognitive Software. And Python has been a choice for most of the developers in these streams.

[Reply](#)**Nihal Singh**

11 Apr 19 at 7:48

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.

[Reply](#)**Helena Orstem**

5 May 19 at 3:58

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.

Reply



medicamentos baratos on-line

30 Jun 19 at 11:37

What's up friends, its fantastic article on the topic of cultureand completely defined, keep it up all the time.

Reply



χάπια χαμηλών τιμών

30 Jun 19 at 12:09

Very good article. I'm facing some of these issues as well..

Reply



Mukshi Sharma

30 Jul 19 at 4:04

In lower-income (non-English speaking) countries, they do not choose Python because many outputs default to the ASCII character set.

Reply



vic

16 Aug 19 at 8:22

I need to explore Python books and start working on it!

Reply

**John Petser**

17 Aug 19 at 1:08

It'd be interesting to collect information to find out how many ongoing with python from prototype to manufacturing. Python would definitely help build MVP, when in creation in which real concurrency things; quite few like Instagram have constructed solid python established architectures.

[Reply](#)**Cliparts**

9 Oct 19 at 6:36

In lower-income (non-English speaking) countries, they do not choose Python because many outputs default to the ASCII character set.

[Hassan](#)[Reply](#)**Water Flosser**

21 Oct 19 at 4:30

In lower-income (non-English speaking) countries, they do not choose Python because many outputs default to the ASCII character set.

[Reply](#)**Programming Apps**

20 Nov 19 at 10:21

Agree Python is fastest-growing programming language

In last few weeks, I was reading about Python history and got shocked to know that big companies like Netflix, IBM and Dropbox uses Python.

[Reply](#)

**Andrew Iacov**

26 Dec 19 at 9:13

The best all-round language we have? Probably.

Now, personally, I prefer C# to Java as a language, I also prefer WPF to Java's UI toolkits (although JavaFX is very good indeed). However, C#/WPF only runs on Windows, so what if I want to run on something else?

Cross platform options are maybe...

C++/QT, C#/Toolkit of your choice, or maybe stuff like TCL/TK. Or maybe you could do a UI in SDL and have it run on just about anything. You could do a HTML/CSS UI too, you could get that running on most things.

For server stuff, you've got a lot more options as you don't need to worry about UI toolkits, but even then, what cross platform languages are there, which are genuinely better than Java? I'm struggling to come up with one, C of course is The Greatest Language Ever Created By Man, but it's not great at everything, if I was writing a corporate system accessing databases, working with XML, REST services, SSL, give me Java any day.

Java is not a perfect system, but it is very good indeed, as said elsewhere in this question, yes it can go crazy with academic stuff, and be needlessly verbose.

However, if not Java, what? Python, Ruby, PHP etc. all have their place, but as an all-round language, Java really is just head and shoulders over all of them.

Reply

**Joy**

6 Jan 20 at 2:07

Is there anything which can be done using python But can't be done using C++ !!!

Reply

**Ayush**

28 Feb 20 at 1:20

Yeah, it's best.. Along with that, it's not that hard to understand the language.

Reply



Raj Kumar

14 May 20 at 7:11

In the last few weeks, I was reading about Python history and got shocked to know that big companies like Netflix, IBM, and Dropbox uses Python.

Great post!

Reply



Aarhan Sharma

21 May 20 at 2:46

Python would definetly help to build MVP, when in production where actual concurrency matters

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment

This site uses Akismet to reduce spam. [Learn how your comment data is processed](#).



© 2020 All Rights Reserved.
Proudly powered by [Wordpress](#)

STACK OVERFLOW

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

PRODUCTS

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

COMPANY

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

CHANNELS

[Podcast](#)
[Newsletter](#)
[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)