

tecosystems

because technology is just another ecosystem

The RedMonk Programming Language Rankings: June 2016

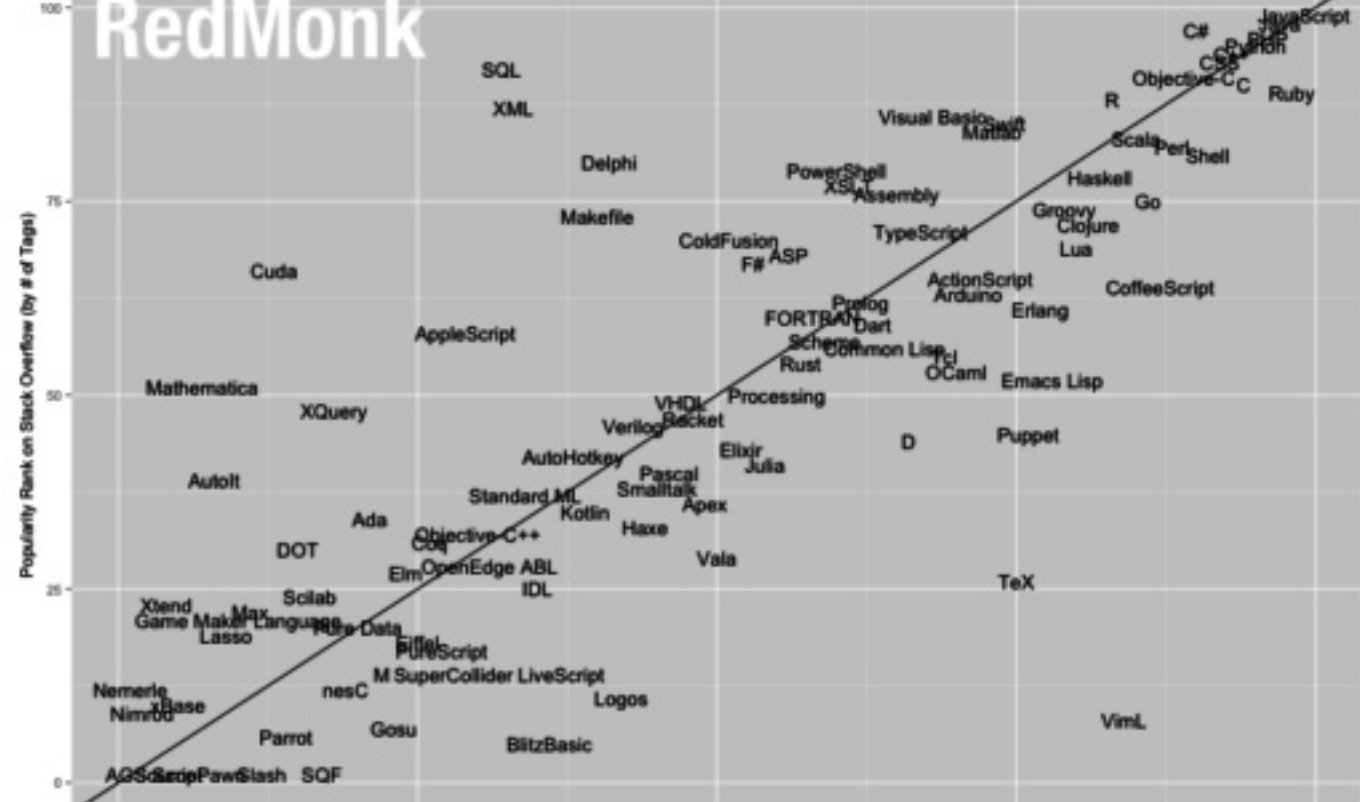
With the spring and summer travel schedule drawing to a close, we finally have had time to sit down and run the numbers collected back in June. As always, aside from the fact that we run our own GitHub rankings now, the process used for our bi-annual programming language rankings remains the same as when Drew Conway and John Myles White first looked at the question [late in 2010](#). We have continued this process, comparing the performance of programming languages relative to one another on GitHub and Stack Overflow twice a year. The idea is not to offer a statistically valid representation of current usage, but rather to correlate language discussion (Stack Overflow) and usage (GitHub) in an effort to extract insights into potential future adoption trends.

With the exception of GitHub's decision to no longer provide language rankings on its Explore page – they are now calculated from the GitHub archive – the rankings are performed in the same manner, meaning that we can compare rankings from run to run, and year to year, with confidence.

Historically, the correlation between how a language ranks on GitHub versus its ranking on Stack Overflow has been strong, but this had been weakening in recent years. From its highs of .78, the correlation was down to .73 this time last year – the lowest recorded. For this run, however, the correlation between the properties is once again robust. As with last quarter's ranking, the correlation between the properties was .77, just shy of its all time mark. This is arguably [noise](#), but we believe the correlation is worth noting at a minimum.

Before we continue, please keep in mind the usual caveats.

- To be included in this analysis, a language must be observable within both GitHub and Stack Overflow.
- No claims are made here that these rankings are representative of general usage more broadly. They are nothing more or less than an examination of the correlation between two populations we believe to be predictive of future use, hence their value.
- There are many potential communities that could be surveyed for this analysis. GitHub and Stack Overflow are used here first because of their size and second because of their public exposure of the data necessary for the analysis. We encourage, however, interested parties to perform their own analyses using other sources.
- All numerical rankings should be taken with a grain of salt. We rank by numbers here strictly for the sake of interest. In general, the numerical ranking is substantially less relevant than the language's tier or grouping. In many cases, one spot on the list is not distinguishable from the next. The separation between language tiers on the plot, however, is generally representative of substantial differences in relative popularity.
- GitHub language rankings are based on raw lines of code, which means that repositories written in a given language that include a greater amount of code in a second language (e.g. JavaScript) will be read as the latter rather than the former.
- In addition, the further down the rankings one goes, the less data available to rank languages by. Beyond the top tiers of languages, depending on the snapshot, the amount of data to assess is minute, and the actual placement of languages becomes less reliable the further down the list one proceeds.



(click to embiggen the chart)

Besides the above plot, which can be difficult to parse even at full size, we offer the following numerical rankings. As will be observed, this run produced several ties which are reflected below (they are listed out here alphabetically rather than consolidated as ties because the latter approach led to [misunderstandings](#)). Note that this is actually a list of the Top 21 languages, not Top 20, because of said ties.

- JavaScript
- Java
- PHP
- Python
- C#
- C++
- Ruby
- CSS
- C
- Objective-C
- Shell
- R
- Perl
- Scala
- Go
- Haskell
- Swift
- Matlab
- Visual Basic
- Clojure
- Groovy

JavaScript retains its position atop the rankings for yet another quarter, as do Java and PHP in their second and third positions respectively. There is no movement, in fact, among languages ranked within our Top 10. The positions have solidified, and it's becoming apparent that it will take a serious push – or crisis – to significantly alter the dynamics of the top tier absent minor and statistically irrelevant drifts from quarter to quarter. It may or may not suggest that fragmentation is beginning to slow, but that's an analysis outside the scope of these rankings.

We do have movement outside of the Top 10, however. Here they are in no alphabetical order.

- Elixir:** Elixir jumped again this quarter, but to a smaller degree (2 spots) than last (6) run. Its trajectory and functional appeal make it a language to watch, but whether or not Elixir can sustain this momentum is the important question. As even very popular languages like Swift have proven, the difficulty of growth is proportional to the rankings themselves – as one rises, so does the other. It's also worth noting that Erlang has not seen a bounce from Elixir; it was static this period, holding at 26.
- Julia:** Julia's growth has always been slow, but this is the first period in a number of quarters where Julia actually slid. Having moved up to #51 last quarter, it slid back to #52 for this run. This is not particularly surprising, as the language is not currently demonstrating the traction, visibility and enthusiasm characteristic of faster adoption rates. We'll watch over the next quarter or two to see whether Julia can resume its climb, or whether it has stalled in a manner similar to CoffeeScript.
- R:** Out of all the back half of the Top 20 languages, R has shown the most consistent upwards movement over time. From its position of 17 back in 2012, it has made steady gains over time, but had seemed to stall at 13 having stuck there for three consecutive quarters. This time around, however, R took over #12 from Perl which in turn dropped to #13. There's still an enormous amount of Perl in circulation, but the fact that the more specialized R has unseated the language once considered the glue of the web says as much about Perl as it does about R. Which is irrelevant to R advocates, of course. Whatever the cause, R's relatively unique Top 20 path is one for fans of the language to cheer.
- Rust:** Interestingly, given that the past two quarters have anecdotally seen an uptick in Rust discussion, the language actually followed Julia's lead and gave up one spot in the rankings this quarter. From a big picture standpoint, this is not particularly problematic, given that individual ranks should be taken with a grain of salt always, particularly so the further down the rankings a given language sits. That said, upward trajectories are preferable to the opposite, even if the actual rankings themselves are not to be obsessed over. Like Julia, it will be interesting to see whether or not Rust will gain next quarter or if it has instead plateaued.
- Swift:** Swift at this point has become the canonical example for the inertia of incumbent languages. Have followed an unprecedented growth trajectory since its introduction, this run is the first in which Swift has not gained but merely held its position of #17. In Swift's defense, it at least performed better than the language directly ahead of it, Haskell, which fell out of a tie with Go for 15th place into #16. But it's clear that further gains for Swift will not come easily, and will instead be the product of widespread usage across an array of communities. As discussed in the last iteration of these rankings, Swift has opened up new avenues for growth beyond iOS development via its release as open source software and the embrace of third parties like IBM or Perfect, but these have yet to yield gains in new discussion or code sufficient to propel it forward in these rankings. We'll be watching for signs of this type of new growth closely.
- TypeScript:** Outside of Go or Swift, the fastest growing language we've observed in recent years is TypeScript. The Microsoft-backed JavaScript superset and [Angular 2 foundation](#) has made significant gains for the second consecutive quarter, jumping from #31 to #26. That was the biggest single change in any Top 30 language, and the second largest jump overall (Standard ML, 7 spots). At #26, in fact, TypeScript is now tied with Erlang, one spot behind Powershell and four behind CoffeeScript, which is just outside the Top 20. The question facing the language isn't whether it can grow, but whether it has the momentum to crack the Top 20 in the next two to three quarters, leapfrogging the likes of CoffeeScript and Lua in the process.

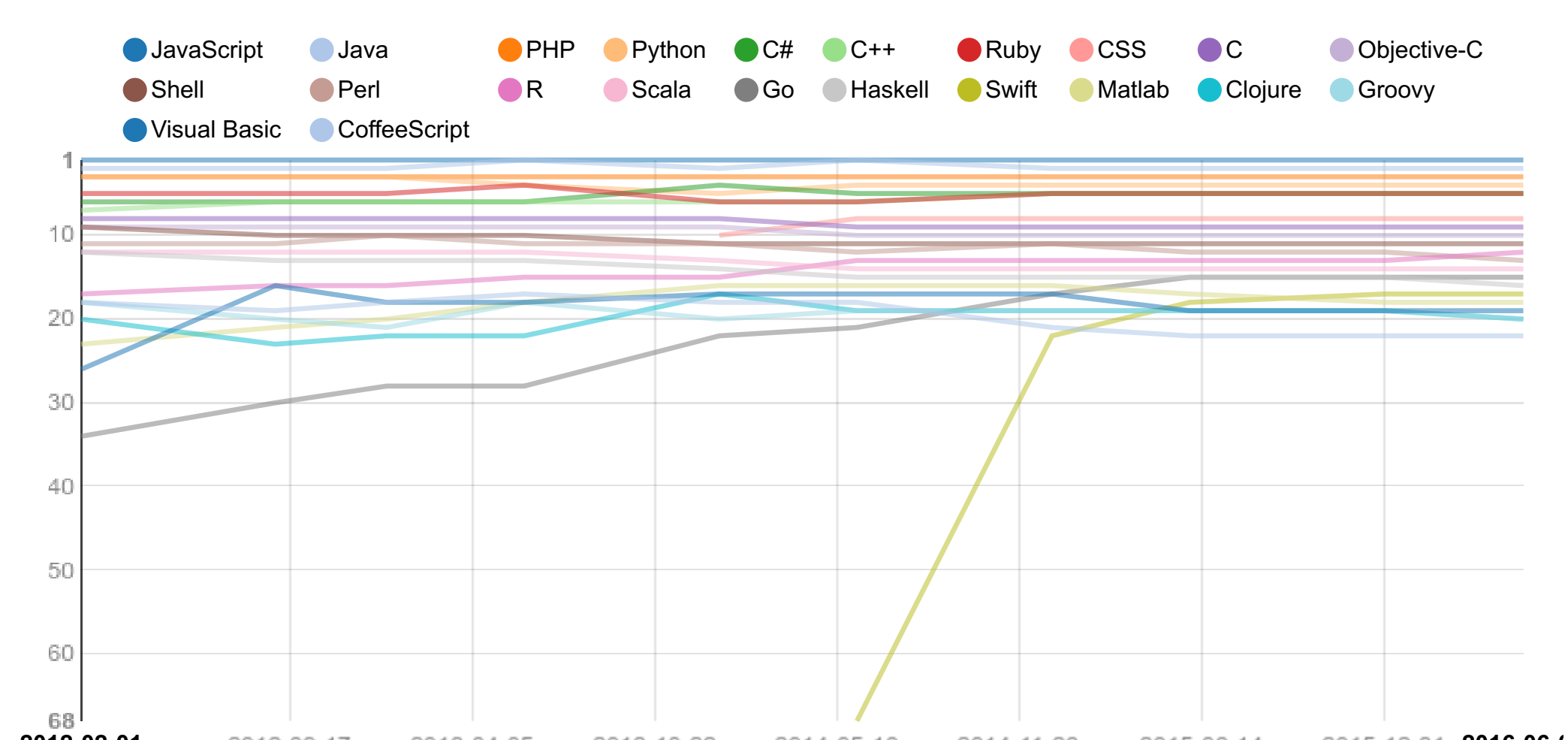
The Historical Rankings

As we did last quarter, this visualization will allow you to dynamically select or deselect languages at will, tracing their individual rankings back to the first runs of this exercise.

A few notes:

- This is not a complete ranking of all the languages we survey. It includes only languages that are currently or have been at one time in the Top 20.
- This graphic is interactive, and allows you to select as many or as few languages as you prefer. Just click on the language in the legend to toggle them on or off. This is helpful because Swift fundamentally breaks any visual depiction of growth: de-select it and the chart becomes much more readable.
- The visualization here, courtesy of Ramnath Vaidyanathan's rCharts package, is brand new and hasn't been extensively tested. Mobile may or may not work, and given the hoops we had to jump through to host a D3-based visualization on a self-hosted WordPress instance, it's likely that some browsers won't support the visualization, HTTPS will break it, etc. We'll work on all of that, and do let us know if you have problems, but we wanted to share at least the preliminary working copy as soon as we were able.

With that, we hope you enjoy this visual depiction of the Historical Programming Language Rankings.

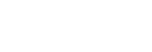


The Net

In general, these rankings are experiencing less volatility over time. The Top 10 in particular is fairly static, and even within the Top 20 movement is becoming more limited. It may be, as mentioned above, that we're at or near peak fragmentation, and that the Cambrian explosion of programming languages is leading to a predictable period of consolidation. We'll look at this question in more detail shortly.

Even if that is the case, however, it is not true that there are no interesting trends to watch in the programming language landscape. R's continued ascent is interesting, particularly following the acquisition of Revolution Analytics by Microsoft. Another language with ties to Microsoft, TypeScript's ascent is as notable as it is surprising, and whether Swift can pass Haskell and potentially even Go will be fascinating to watch. In the last few spots in the Top 20, meanwhile, there are suggestions that change is coming: Clojure, Groovy and Haskell all fell back this quarter.

We'll be back with you in two quarters to assess these and other questions.



8 Comments

Categories: [Programming Languages](#)
By [sogrady](#) — July 20, 2016 at 2:54 pm

8 Responses

[Comment Feed](#)

Delphi and Pascal are separate entries in the ranking, but what defines the difference between them?

The two major Pascal compilers (Delphi and Free Pascal) are both Object Pascal compilers. So shouldn't the Delphi and Pascal entries be merged into a single Object Pascal entry?

Jim T — July 20, 2016 @ 10:08 pm — [Reply](#)

Why do you rank after Stack Overflow tags? In my humbled opinion that's an extraordinary bad measure as it (1) shows how complicated a language is (i.e. how much help you need), (2) many languages have specialized ways of getting help (forums, IRC, issues), (3) in some languages like JavaScript many beginners start & thus ask many dumb questions that experienced users in professional languages will never bother to ask, (4) script languages due to their nature have shorter & more questions, (5) languages that have been out there for a while with extensive documentation are not represented accordingly.

I do understand that you want to find a way to measure "popularity", but please don't use StackOverflow community tag – it's its_not_a representative figure.

Secondly using lines of code is yet another bad measure, because in a good language I can achieve the same with 10x less lines of code.

Versalis — July 21, 2016 @ 6:45 am — [Reply](#)

"We do have movement outside of the Top 10, however. Here they are in no alphabetical order." Yet, they are in alphabetical order...

It looks like maybe you started to write "in no order" and changed your mind, choosing alphabetical instead, but forgot to take out the word "no".

Mike — July 21, 2016 @ 7:42 am — [Reply](#)

Do we consider CSS a programming language? I use it and I like it but I do not think it can be used for anything other than styles for HTML web pages. Please correct me if I am wrong.

CTrom — July 21, 2016 @ 10:19 am — [Reply](#)

One thing which may not be factored into your calculations would lead to stack overflow: TeX has its own stack overflow [@ tex.stackexchange.com](#) which is where you'll find the stack overflow activity, rather than in the main stack overflow.

David T. Macknet — July 21, 2016 @ 10:57 am — [Reply](#)

Sorry, did I miss a link to a fuller lineup? Is there any data on languages below 21? Just looking for the trends on a language I'm interested in that clearly, not many people are using.

Brian — July 21, 2016 @ 11:59 am — [Reply](#)

Interesting to see GO move up in the rankings once again. Year after year, the steady rise of GO is testament to the expressive power of a language that is forming the corner stone of Cloud infrastructure, whether it be Red Hat's OpenShift, Docker, IBM Cloud-Foundry/Bluemix or the industry wide migration trends away from Python/Java/.NET when it comes to greenfield and non-legacy focused DevOps, in developing cloud/native applications and databases like the opensource ACID compliant CockroachDB being built with GO instead if C/C++.

Alexander Manley — July 21, 2016 @ 5:33 pm — [Reply](#)

Leave a Reply

Some HTML is OK

Name (required)

Email (required, but never shared)

Web

or, reply to this post via [trackback](#).

Continuing the Discussion

[...] Source: [RedMonk](#) [...]

[JavaScript Is Still The King — Top 21 Programming Languages Of 2016](#) — July 21, 2016 @ 4:46 am

« [Hark Episode 3, "Getting Medieval on You": Guest, KellyAnn Fitzpatrick](#)

About

Hi, I'm Stephen O'Grady. I live in Maine, but [travel](#) a lot.

I helped found RedMonk in 2002, and I was born and raised a Red Sox fan.

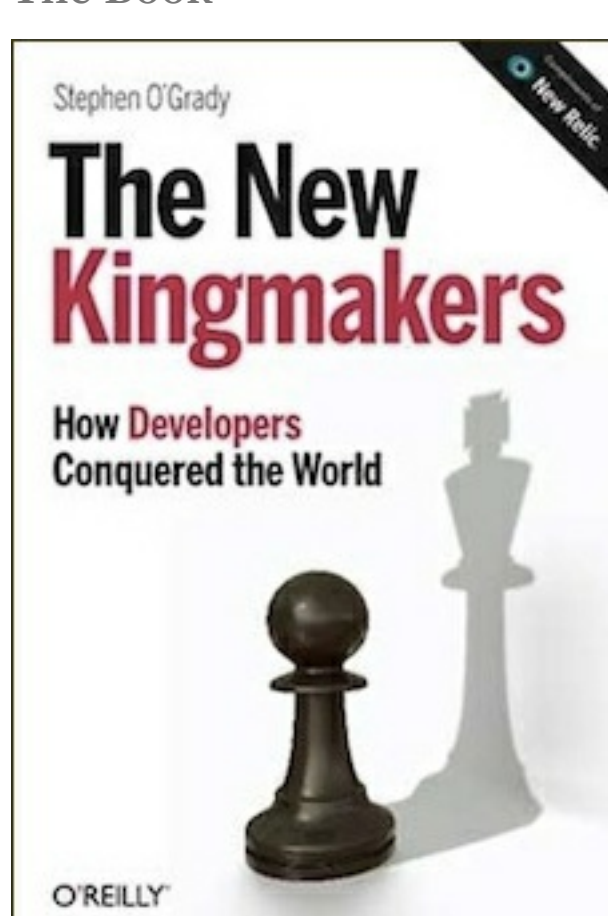
My job is to help companies understand developers better, and to help developers, period. There's more bio stuff [here](#).

[Follow @sogrady](#)

Newsletter

Weekly links, commentary and recommendations. [Subscribe here.](#)

The Book



Search

Search for:

Recent Posts

- [The RedMonk Programming Language Rankings: June 2016](#)
- [Hark Episode 3, "Getting Medieval on You": Guest, KellyAnn Fitzpatrick](#)
- [There and Back Again: The MongoDB Cloud Story](#)
- [Why LinkedIn and Microsoft Isn't Crazy](#)
- [Hark Episode 2, "The Software Paradox": Guest, Kent Beck](#)

Recent Comments

- [Alexander Manley](#) on [The RedMonk Programming Language Rankings: June 2016](#)
- [Report: JavaScript is still the top-ranked programming language - SD Times](#) on [The RedMonk Programming Language Rankings: January 2015](#)
- [Brian](#) on [The RedMonk Programming Language Rankings: June 2016](#)
- [David T. Macknet](#) on [The RedMonk Programming Language Rankings: June 2016](#)
- [CTrom](#) on [The RedMonk Programming Language Rankings: June 2016](#)

Categories

Categories

Archives

Archives