

RedMonk Programming Language Rankings (June 2016)

Source: <http://redmonk.com/sogrady/2016/07/20/language-rankings-6-16/>

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 analysis, 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.

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.

