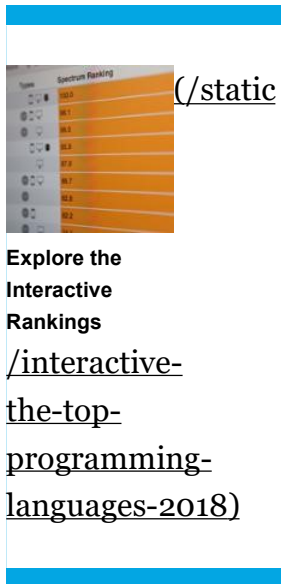


31 Jul 2018 | 15:00 GMT

The 2018 Top Programming Languages

Python extends its lead, and Assembly enters the Top Ten

By **Stephen Cass** (</author/cass-stephen>)



Welcome to *IEEE Spectrum's* [fifth annual interactive ranking of the top programming languages](/static/interactive-the-top-programming-languages-2018) (</static/interactive-the-top-programming-languages-2018>). Because no one can peer over the shoulders of every coder out there, anyone attempting to measure the popularity of computer languages must rely on proxy measures of relative popularity. In our case, this means combining metrics from multiple sources to rank 47 languages. But recognizing that different programmers have different needs and domains of interest, we've chosen not to blend all those metrics up into One Ranking to Rule Them All.























Instead, our interactive app lets you choose how these metrics are weighted when they are combined, so you can put an emphasis on what matters to you. (There's a [detailed description of our methods and sources](/static/ieee-top-programming-languages-2018-methods) (</static/ieee-top-programming-languages-2018-methods>) available.) We do include a default weighting, tuned to the interests of a typical IEEE member, and we offer some other presets that focus on things like what's *au courant* for open source projects. You can also apply filters that exclude languages primarily used in areas that aren't of interest to you, such as embedded or desktop environments. And you can see how things have changed by making comparisons with earlier years.

So what are the Top Ten Languages of 2018, as ranked for the typical IEEE member and *Spectrum* reader?

Python has tightened its grip on the No. 1 spot. Last year it came out on top by just barely beating out C, with

Python's score of 100 to C's 99.7. But this year, there's a wider gap between first and second place, with C++ coming in at 98.4 for the No. 2 slot (last year, Java had come third with a score of 99.4, while this year its fallen to 4th place with a score of 97.5). C has fallen to third place, with a score of 98.2.

Why is Python continuing to gain programmer mindshare? Two other changes in the Top Programming Languages may give a hint as to why.

Language Rank	Types	Spectrum Ranking
1. Python	  	100.0
2. C++	  	98.4
3. C	  	98.2
4. Java	  	97.5
5. C#	  	89.8
6. PHP		85.4
7. R		83.3
8. JavaScript	 	82.8
9. Go	 	76.7
10. Assembly		74.5

First, Python is now listed as an embedded language. Previously, writing for embedded applications tilted heavily toward compiled languages, to avoid the overhead of evaluating code on the fly on machines with limited processing power and memory. But while Moore's Law may be fading (</static/special-report-50-years-of-moores-law>), it's not dead yet. Many modern microcontrollers now have more than enough power to host a Python interpreter. A nice aspect

of using Python this way is that it is very handy in certain applications (</geek-life/hands-on/build-an-illuminated-halloween-costume-with-the-wearable-gemma-mo-microcontroller>) to play with attached hardware via an interactive prompt or dynamically reload scripts on the fly. Growing into a new domain can only help boost Python's popularity.

Another reason for Python's increasing popularity may be seen in R's small decline. R peaked at No. 5 in 2016, dropped to No. 6 last year, and is now in seventh place. R is a language specialized for handling statistics and big data. As the interest in large data sets has increasingly turned to their applications in machine learning, the existence of high-quality Python libraries for both statistics (<https://www.scipy.org/>) and machine learning (<https://keras.io/>) may be making flexible Python a more attractive jumping-off point than the more specialized R.

Looking at the Trending preset, designed to weight the metrics to emphasize languages that are growing quickly, we see that Google's Go has risen from No. 7 to No. 5. But perhaps the biggest mover is Scala, rising from No. 15 to No. 8. Scala was created to be an improvement over Java, so perhaps Java's drop in the default ranking owes something to the upward trend for Scala.

Surprisingly, ActionScript (<https://www.adobe.com/devnet/actionsript.html>) has eked out a small bump up in the rankings. Last year it came in dead last with a ranking of 0.0, so we were all set to eliminate it. However, it's managed to come back into second-to-last place with a score of 1.6, while Forth, once a workhorse of the embedded world takes the goose eggs. I would be sad to see Forth go, as it's one of my personal favorites, but if it comes in a zero again next year, it'll be axed.