GEEK ON

Not just “IT coolies”: An Indian computer scientist helped build an alternative programming language in Bengaluru

Maria Thomas  |  April 20, 2017  |  Quartz India

Anywhere, anytime. (Unsplash/Hannah Wei)

You are reading Quartz India. To switch to the global edition, click here.
and never come back.

And yet, there are plenty of Indians who go on to do great things by taking a completely different route. Viral Shah, CEO of Julia Computing and co-creator of the Julia programming language, is one of them.

Julia is a free, open-source computer programming language that is gradually becoming a popular alternative to more established languages such as MATLAB and Python. Envisioned as a way to avoid the difficulties of using slower, older languages for today’s more advanced analytical applications, without compromising on ease of use, Julia has found fans around the world, drawing a sizable community of users since its launch in 2012.

And its success comes in part from Shah’s efforts not in Silicon Valley but right here in Bengaluru.

The 37-year-old grew up in Mumbai and went to the lesser-known Padmabhushan Vasantdada Patil Pratishthan College of Engineering in one of the suburbs of the city, where he studied computer engineering. As a student, Shah remembers neglecting his coursework and focusing on his own research instead. When it came to higher studies, his poor grades meant that only one school accepted him: the University of California, Santa Barbara (UCSB), where he went on to get a PhD in computer science.
Julia: Viral Shah, an Indian computer scientist built a new programming language from Bengaluru — Quartz

[Image]

Julia Computing’s Viral Shah.

You are reading Quartz India. To switch to the global edition, click here.
Years later, along with fellow UCSB student Stefan Karpinski and Jeff Bezanson and Alan Edelman from the Massachusetts Institute of Technology (MIT), Shah began working on the language that would become Julia.

“I usually find it disappointing when people say that everything is done in the US,” he told Quartz, and Julia’s growing success proves that innovation can come from India, too.

**Best of both worlds**

Julia’s story begins in 2009 when Shah and its other co-creators were looking to simplify the coding process. For long, computer programmers, applied mathematicians, and data scientists could either use higher-level languages like MATLAB or Python, or brave the C or Java languages that are better suited for advanced applications but a lot harder to use, especially for those who aren’t programmers by training.

“So what ends up happening is…the two-language problem, which basically says that you write your algorithm first in a high-level language and then rewrite it in a low-level language for performance, or taking it live. And so you end up writing everything twice,” Shah explained. “If we had one language that was simultaneously easy to use and fast, you could just unleash innovation across the world.”

And that’s what the team set out to create, largely over email because 2009 was also the year Shah moved to Bengaluru to work on India’s Aadhaar project with Infosys co-founder Nandan Nilekani.

By 2012, the language was ready to be revealed, and on Valentine’s Day the team celebrated its public launch with a blog post explaining its mission and inviting
contributors to pitch in and improve the original code. Shah says the post went viral on Reddit and Hackernews, sparking interest around the world.

Within a year, the team released the first version of the language for free with help from over a 100 open-source contributors. Since then, the Julia community has grown even bigger, with an estimated 250,000 users around the world today. This includes a thriving group of over 400 in Bengaluru itself.

Shah says the language has been roughly doubling its user-base every nine months. Moreover, it has attracted some high-profile clients, from investment manager BlackRock, which uses it for time-series analytics, to the British insurer Aviva, which uses it for risk calculations. And in 2015, the Federal Reserve Bank of New York used Julia to make models of the US economy, noting that the language made model estimation “about 10 times faster” than before.

Encouraged by the response, Shah and the language’s other co-founders established Julia Computing in 2015 to provide paid support, training, and consulting services to clients, though Julia itself remains free to use.

While the language has faced some criticism—blame certain bugs—Shah is hoping it will evolve and improve, eventually hitting a user-base of at least a million programmers in the next year or two.

The other key goal is to eventually have every college and university teaching and using it. Currently, some 300-odd institutions around the world, including MIT, Stanford, and Cornell, have already taught courses with it. MIT has a Julia lab dedicated to the research and development of the language, and the fourth annual JuliaCon conference will take place at the University of California, Berkeley this year.
Amazing India

Till now, only a handful of Indian universities, notably IIT-Indore, have been teaching Julia. IIT-Indore conducted one course in numerical linear algebra methods last year that was taught entirely in the new language. And the idea to do so came from Ivan Slapničar, a visiting professor who had learnt Julia at MIT from Edelman himself.

For Slapničar, a professor at the University of Split in Croatia, it was the philosophy and implementation of the language that appealed to him, with its ease of use and speed improving on the drawbacks of existing languages. However, he noted that changes between versions of the language could make it hard for general users to keep up, and that more testing needs to be done to make sure everything works perfectly, meaning that there’s a long road ahead before Julia can gain wider acceptance in Indian universities.

“My feeling is that it still depends mostly on the enthusiasm of the professor,” Slapničar said in an email, noting that the first obligation is to teach students the most popular languages, such as MATLAB and Python.

And therein lies a particularly local problem:

“...The Indian college curriculum is very centrally-driven, so professors typically can’t make their own choices, unlike in a US university where the professor teaching the class can decide what he’s going to use,” Shah said.

Having experienced the Indian education system, Shah believes its approach can sometimes stand in the way of students being exposed to new ideas.

“I think the basic point is that it’s too coursework-focused and (has) nothing in
that are controlled by a company. So, you’re teaching your students tools which they can’t afford to buy after they graduate.”

But despite the rigidity of the system in India, Shah still doesn’t believe you need to go anywhere special to innovate—not to an IIT, not even outside the country.

“People say that Bangalore (Bengaluru) only has IT coolies and no one does anything meaningful,” Shah said, pointing out that nothing could be further from the truth.

*We welcome your comments at ideas.india@qz.com.*