COMMUNICATIONS
OF THE
ACM

Centrum Wiskunde and

Informatica

Search

ACM NEWS

Python at 25

By Bennie Mols
February 24, 2015
Comments

VIEW AS: SHARE:

Twenty-five years after creating Python, Guido van Rossum still finds the programming language "easy to learn and easy to use, and it works in a very interactive way."

Credit: Bennie Mols

For many years, Python has appeared on the list of the 10 most-utilized computer languages in the world. Companies like Dropbox, Google, Mozilla, and Walt Disney Animation Studios are large users of Python, as are banks, universities and institutions such as the U.S. National Aeronautics and Space Administration (NASA) and CERN (the European Organization for Nuclear Research).

Dutch software engineer Guido van Rossum created the language in 1990 while working at the Centre for Mathematics and Computer Science (CWI) in Amsterdam, The Netherlands. He named the language after the British comedy series *Monty Python's Flying Circus.*

In 1995, van Rossum moved from the Netherlands to the U.S., where he worked as a software engineer at the U.S. National Institute for Standards and Technology (NIST), the Corporation for National Research Initiatives (CNRI), and Google. Since December 2012, he has been worked for the cloud service Dropbox in San Francisco. In 2001, he received the Free Software Foundation's Advancement of Free Software Award, and in 2006 he was recognized by *ACM* as a Distinguished Engineer.

I visited van Rossum at his home in Belmont, CA, where he patiently spoke with me for two hours about the development of Python and his passion for software engineering.

How do you explain the success of Python?

Python is easy to learn and easy to use, and it works in a very interactive way. Computer programmers can get new results very quickly. It's also a great tool for scientists who work interactively with data. Furthermore, it has been very important that from the beginning, Python has been available according to the open source philosophy, even before the term "open source" was invented. Python users feel part of a community and are very active in making improvements.

The Python user community gave you the title "Benevolent Dictator for Life." Does that mean that in the end, it is you who decides about changes in Python?

That term is, of course, a bit of a joke, in the line with the Monty Python type of humor. In the reality of Python, I have very little real power. I am intensely involved in discussions about the language, but basically I am sending e-mails while sitting behind my computer; that's the only thing I can do. I only have the power of persuasion. Only when it comes to syntactic modifications of Python can I make decisions.

MORE NEWS & OPINIONS

Haptic Holograms Let You Touch the Void in VR
New Scientist

HPC for Undergraduates
Jeffrey Hollingsworth

Will Autocomplete Make You Too Predictable?
BBC News

ACM RESOURCES

Oracle Fundamentals (Second Edition)
Courses

What are you most proud of?

That I have created a computer language that, after 25 years, is still successful. Recently I went back to a Python version from 1991, the first international release; it still looks very much like today's Python. The basic ideas are still the same. Of course, many things have been added over the years, thanks to the active open source development. I have always highly enjoyed interaction with other programmers to try and solve programming problems together.

Now you work at Dropbox, a young company with a lot of young programmers. How is your interaction with this new generation?

I love to build software with our bright young programmers. I still learn about software development and how to solve problems. However, sometimes I do feel like a representative of an older generation; I have so many years of experience that hardly anybody at Dropbox shares with me.

Some of my experience has become irrelevant. For example, I still do remember using punch card machines in the 70's.

On the other hand, other parts of my experience are of added value. Sometimes I can say, "this is a problem that we were already struggling with 30 years ago." I can put programming in a wider perspective.

The number of bugs per thousand lines of code has not gone down over the decades, and the ideal size of a software development team has also not changed; a bit, depending on what exactly you are building. It's, let's say, between five and 20 people."

Software is becoming ever more important in our world. How do you look at the state of today's software?

I cannot talk in general terms, only about specific applications.

Okay. Let's take the autonomous car as an example.

The near-future is one in which the car can do more and more things autonomously, from staying in one lane to parking. However, the completely autonomous car, that takes you everywhere like a taxi, is still far away. There are many situations in which the self-driving car does not perform so well, and that is often not mentioned. Humans are still much better when it comes to vision, and especially to interpreting what they are seeing. Computer vision has to overcome many problems, like changes in light conditions. I myself would prefer a car that helps me where it is needed, instead of a car that takes over everything.

You live and work in Silicon Valley. Do you already see the next big thing arising?

Well, I am not much of a visionary. To give you an example: I was taken completely by surprise by the invention of the World Wide Web in 1991. But I do think that software is becoming more and more a part of us as human beings. When you drive a car or ride a bike, you can do that for a large part unconsciously; the vehicle has become a part of you. The same is true for mobile phones. The end of software becoming an integral part of ourselves is not in sight by far."

Bennie Mols is a science and technology writer based in Amsterdam, the Netherlands

---

No entries found

Comment on this article

Signed comments submitted to this site are moderated and will appear if they are relevant to the topic and not abusive. Your comment will appear with your username if published. View our policy on comments