Guido van Rossum: The Early Years of Python

Charles Severance, University of Michigan

Guido van Rossum discusses the initial development of Python, which has increasingly become the programming language of choice for many scientific fields due to its extensibility and ease of use.

Combining simplicity and ease of learning with powerful capabilities and strong performance, Python is one of the most popular programming languages in the world. Created by Guido van Rossum in 1989, Python is the leading application language in important emerging areas such as natural language processing and big data. It fills a very important gap between high-level applications like spreadsheets and statistical analysis packages and systems-oriented languages like C, C++, and Java. You can see my full interview with Guido at www.computer.org/computingconversations.

A NEW OPERATING SYSTEM

In the late 1980s, Guido was working at Centrum Wiskunde & Informatica (CWI), a mathematical and computer science research center in Amsterdam, developing a distributed operating system called Amoeba. Amoeba intended to make a network of computers appear as a single computer using a distributed kernel:

> *We wanted it to be self-hosting, and in order to do that, we realized we needed a large amount of user-level tools like an editor, a mail program, a login utility, or a backup tool. Because the file system model on Amoeba was very different from those on Unix systems, we couldn't use an existing suite of Unix utilities. We had a small team of people working on those tools, but it was very slow-going writing it all in C.*

Guido wondered if there might be a quicker way to get all of the OS utilities built:

> *I had this idea that given how much time we had available for Amoeba, I could actually build a whole new language, design and implement it from scratch, and then use it to implement our suite of tools and still*

0018-9162/15/$31.00 © 2015 IEEE

EDITOR CHARLES SEVERANCE
University of Michigan, csev@umich.edu

> *be ahead of the game compared to a situation where we would have just clunked on writing the things we wanted to write in C.*

For an earlier project, Guido had worked on a language called ABC:

> *I wondered if the ABC language would be a much better language to write these utility tools for Amoeba. But ABC was very high level and abstract, and it wasn't well suited to talking to servers, file systems, and processes. In an alternate universe, ABC could have become the language of spreadsheets, as it was very good for talking about a user's data and doing all sorts of clever stuff using general-purpose data structures like lists and dictionaries. ABC also had nice code structuring devices, like a few simple statements that could be combined to create other constructs.*

Over a long holiday break in December 1989, Guido started developing an ABC-like language that could talk to the OS and would be suitable for quickly developing OS utilities for Amoeba. He named his nascent project Python, taking inspiration from the *Monty Python's Flying Circus* television program. After the holiday break, he continued to evolve the new language in his spare time:

> *For three months I did my day job, and at night and whenever I got a chance I kept working on Python. After three months I was to the point where I could tell people, "Look here, this is what I built." It had an interactive interpreter loop, so the first demos were all, "Let's assign an expression to a variable and print it back," or "Let's define a small function and call it," or "Let's put some things in an array and iterate over the array."*

Though Python wasn't yet ready to be used to develop Amoeba OS utilities, the language was very appealing for programmers who were tired of writing C programs or Unix shell programs for various tasks:

> *My two office mates were almost instantly taken with [Python] and started helping out. A few others within the institute were also excited about Python. We didn't use it on Amoeba right away because it wasn't mature enough to actually develop the system utilities that we wanted. But it worked well on our Unix system, and people outside my department at CWI started using it because it was fun and productive to use.*

After a year of development and use within CWI, Guido and his colleagues decided that Python might have a broader use and so decided to release it as free software. This was before the term "open source" was coined, so they simply looked to the MIT X-11 license as an example of how to release free software. But they needed permission from CWI management to release the product. Guido's manager sent him to the legal affairs department:

> *She asked me questions like, "Did someone pay for this to be developed?," and my answer was "No, I started this all on my own time as part of some research projects and my manager's fine with it." So she said, "Sure, go ahead," and we did it. That was an incredibly lucky stroke.*

Python is the leading application language in important emerging areas such as natural language processing and big data.

The first version of Python came out in February 1991 and was distributed using the alt.sources newsgroup on Usenet:

> *I immediately started getting useful, positive feedback from people who picked up Python from Usenet, and we quickly got into a routine of doing new Python releases. There were the obvious improvements to the language, and the library, and bug fixes. An important category of contributions were ports where people had different architectures and different compilers since the Unix world was much less homogenous at the time. There were a lot of small Unix vendors that had their own compilers or their own hardware, all sorts of incompatibilities.*

From the initial release of Python through the early 1990s, the size of the Python community grew and numerous organizations started depending on the

See **www.computer.org/computer-multimedia** for multimedia content related to this article.

language. With broadening adoption, there was a concern among users that "Guido might get run over by a bus." Some of the adopters were US government agencies that wanted to help grow and stabilize the Python community.

> *I got an invitation from NIST [National Institute of Standards and Technology] to come to the United States for a couple of months. We organized and hosted the first Python workshop at NIST in Gaithersburg, Maryland, in November 1994. Through the Python workshop I met people from the Corporation for National Research Initiatives (CNRI), and they offered me a job working on Python. I went back to the Netherlands for a few months and then from 1995 to 2000 I worked in the US in northern Virginia at CNRI.*

Now Guido could focus on building Python and evolving the user community with solid support from CNRI. During the late 1990s, Python moved through a series of 1.x releases:

> *When I started at CNRI, Python 1.3 was about to be released, and then while I was there we released several subsequent versions leading up to 1.5.2, which remained sort of the gold standard of Python for a long time afterward.*

The growth of the Python community from its creation in December 1989 and the maturation of Python 1.x in the late 1990s laid the groundwork for the even broader expansion of Python 2.x and now 3.x. It's an excellent example of organizations like CWI, NIST, and CNRI making investments in an open source "commons," leading to significant positive value in computing. C

CHARLES SEVERANCE, Computing Conversations column editor and *Computer*'s multimedia editor, is a clinical associate professor and teaches in the School of Information at the University of Michigan. Follow him on Twitter @drchuck or contact him at csev@umich.edu.