Esther Shein

# Python for Beginners

*A survey found the language in use in introductory programming classes in the top U.S. computer science schools.*



THE WAY TAYLOR POULO sees it, learning to code in Python is comparable "to learning Latin and romantic languages." Once someone grasps the logic behind Python, the concepts can be more easily transferred to other languages, maintains Poulos, a senior majoring in industrial engineering at the Georgia Institute of Technology (Georgia Tech). "Once you get comfortable thinking in a different type of logic and using different words, it's much more comfortable to learn new things," she says, adding that she was required to take three computer science classes at Georgia Tech, all in Python. "Python did that."

Python, an open source scripting language, has become the most popular introductory teaching language at top U.S. universities—Georgia Tech among them—according to a recent survey by Philip Guo, an assistant professor of computer science at the University of Rochester. Guo decided to conduct the research after noticing anecdotally over the past few years that Python was replacing languages such as Java as the de facto introduction to programming class in more and more computer science classes at universities around the country.

Because it is a scripting language, Python automates tasks that would otherwise need to be performed manually. Java and C++ also are popular and widely used. The main difference is that Python programs tend to run slower than Java programs, but they take significantly less time to develop, according to the Python Software Foundation. Python programs also tend to be shorter than equivalent programs written in Java because of "Python's built-in high-level data types and its dynamic typing," the Foundation notes. While the same is true of C++, Python code is generally one-fifth to one-tenth the length of equivalent C++ code, and "Anecdotal evidence suggests that one

Python programmer can finish in two months what two C++ programmers can't complete in a year," the Foundation's website states.

During the summer of 2014, Guo went to the websites of the top 39 U.S. schools for computer science as ranked by *U.S. News & World Report* in 2014, and collected as much data as he could from looking at their introductory computer science courses. He stopped at 39, he explains, because there was an eight-way tie for 40 and "we had to stop somewhere." At schools including the Massachusetts Institute of Technology (MIT), Carnegie Mellon University, and the University of California, Berkeley, Python emerged as the leading language to teach novices (the full list, along with Guo's blog on the topic, can be found at at http://bit.ly/W0vtox).

Proponents say it is no surprise Python has become the most popular teaching language in colleges, because compared to programs like Java, it is easier to learn and to use to write programs that do practical things with very little code.

With Python, "There's very little

---

**In contrast to Java, Python makes more sense for people who are writing small programs.**

---

overhead in getting to the point where people can start to write interesting programs; the syntax is pretty straightforward," observes John Guttag, professor of electrical engineering and computer science at MIT, and the author of several books, including one about learning to program in Python. In contrast to Java, which has a "fairly complicated syntax and fairly complicated static semantics," Python makes more sense for people who are writing small programs, he says. Java is designed to support people writing large, "industrial-quality" programs containing thousands of lines of code, says Guttag, who teaches one of two introductory courses offered by his department.

Another reason Guttag believes more colleges are using Python as an introductory programming language is that it has "a very large set of highly useful libraries that have been built over the years that support things … that are easy to use from language proper, and that makes Python a particularly useful language for scientists and engineers who want to take advantage of those libraries."

Python is also very good for "letting you teach conceptual material without getting in the way," observes Guttag. "So I don't find myself spending all my time explaining Python to the students. I get to spend a lot of time explaining what I think are more long-lived concepts," like algorithmic complexity.

Not everyone agrees Python is the be-all-end-all as an introductory programming language. Shriram Krish-

namurthi, a professor of computer science at Brown University, acknowledges Python has many nice features. "It offers a pleasant syntax, a large set of libraries, and an interaction loop ... all of which are very useful for teaching. Compared to the noise and complexity of Java, it is indeed a very nice step forward." He agrees Python has made people feel more comfortable about exposing programming to a much broader audience of students.

"There are many students I would not dream of teaching Java to that I would happily show Python." That said, however, it does not take long to discover Python's weaknesses, Krishnamurthi notes. Among them are that "Creating non-trivial data structures is onerous, because Python does not provide straightforward means for creating new structured data. You have to understand a bunch of unrelated concepts, like classes, and their onerous syntax and tricky semantics, which greatly reduces the benefit of simplicity that Python was supposed to offer."

Because of this, he believes more and more curricula are ditching the idea of structured data—one of the central concepts in computer science—and doing one of two things: shaping their curriculum to avoid them, or pushing students to encode more-structured data in less-structured formats provided by default in Python.

"This lack of data structuring and classification has a significant nega-

> **"Choosing Python is the modern equivalent of the old adage, 'nobody ever got fired for buying IBM.'"**

tive impact on teaching program design," Krishnamurthi says. "The best program design methods we have right now focus on data-driven design, which derive from the structure of data."

Additionally, Python has limited support for testing, he says. Even though it has professional testing libraries, he says they can be "onerous for beginning students; the language provides no native support for it, which makes the user interface of testing significantly weaker than it should be." Testing is not just a matter of finding bugs, he adds. "It also guides students towards the design of solutions and greatly affects how one views debugging. Thus, another vital design and development methodology is taken away."

Lastly, Python lacks static types, says Krishnamurthi, which "is a central point in teaching programming, and should not be put off for too long. Python offers no good means for

teaching this."

Guo says he got some backlash from older colleagues who do not view Python as a serious programming language, along with comments that it is not as "industrial strength" as other languages. One comment he received after posting his blog on the topic is that Python is a dynamically typed language and "There's fair amount of instructors who prefer statically typed languages, like Java," Guo says. "So they aren't as happy about this new movement."

Mark Guzdial, a professor in the School of Interactive Computing at Georgia Tech, says Guo's research notwithstanding, Java is still the most popular introductory programing language in the U.S. Guo "constrained his search to top U.S. universities," Guzdial says, "and in general, if you look at book sales, Java is still the most common [language taught] and C++ is second."
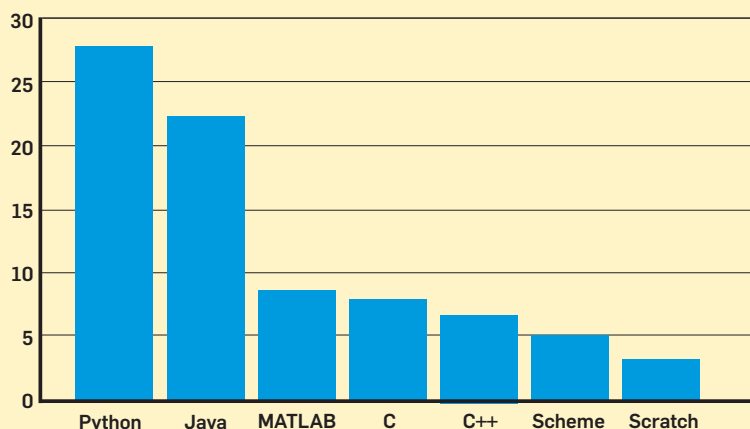
Yet, Guzdial agrees that if someone lacks any prior programming experience, Python is a good language with which to start. "There's a significant amount of evidence that graphical programming languages are easier for people to get started with than textual," he says. "If you understand variables in Scratch, it'll be easier to understand variables or conditionals or loops in Python."

Like Krishnamurthi, Guzdial thinks Python may broaden the scope of people being able to code. "It is easier and more accessible and ... you can get more done in fewer lines of code."

Matt Guthmiller, a sophomore majoring in electrical engineering and computer science at MIT, says he was not terribly surprised by Guo's finding. Guthmiller took his first Python class as a freshman at MIT, but he already knew how to code in C, C++, and JavaScript. "It definitely seems to make a lot of sense as an introductory tool because it's easy to learn, with lots of functionality built in, and you can do things that in other languages you'd have to build yourself, and [in Python] they're provided for you."

He likes that Python allows you summarize a list of data in one line of code, whereas in other languages it would take multiple lines. "You have to think about the order you want to iterate these items and implement

Number of top 39 U.S. computer science departments that use each language to teach introductory courses

| Language | Number |
| --- | --- |
| Python | 28 |
| Java | 22 |
| MATLAB | 8.5 |
| C | 8 |
| C++ | 6.5 |
| Scheme | 5 |
| Scratch | 3 |

Analysis done by Philip Guo (www.pgbovine.net) in July 2014, last updated 2014-07-29

them, so there's a lot more functionality built into one line, so if there's a problem you want to solve in one line of code, you get much closer to solving the problem than in other languages."

Like Guttag, Guthmiller feels the biggest disadvantage of Python is that "the syntax is quite different from most other programming languages," making it trickier to move on to another language once you get all the general concepts down. However, he says, Python's advantages outweigh its disadvantages.

Guthmiller recently used Python to build a controller for a robot to make it follow along a wall, although, generally speaking, his go-to programming language is C++. Python, he says, "gives you a lot of flexibility, and I'm very familiar with it and I am not concerned about having to remember small details."

Abbie Burton, a senior majoring in business at Georgia Tech, was required to take a computer science class and took "Jython," a combination of Java and Python that business students tend to take. She says most engineering students take Python or MATLAB, and she is not surprised by Python's popularity, "I guess because in the real world that's what people use, so they want us to be prepared."

There does appear to be a preference for using Python outside of academia. For the third year in a row, Python was ranked the number one most popular programming language by Codeval, a community of over 24,000 competitive developers, followed by Java, C++, and JavaScript. (http://bit.ly/1vLiuFj).

Guo says he has heard some comments that while easy to learn, Python does not have practical applications in the real world and that most coding is done in MATLAB and other languages. He says it all depends on the domain. "MATLAB is used in a lot of scientific domains. I definitely think it's less practical in terms of getting an industry job, because most industry coding would be in other languages, like Java or JavaScript. So I would agree it might not be the language you'd use in your job."

Guttag says Python is a useful tool for people who do not intend to be computer scientists, because it provides a good foundation for learning how to use computation as part of their work.

"For those non-computer science students," he says, "Python is an excellent choice as the introductory programming language."

Krishnamurthi says Python may be fashionable right now, but he believes it lacks staying power. "Computer science programming education goes in waves of fashion," he says. "Ever since (Blaise) Pascal introduced the idea of 'one programming language for introductory programming education,' the community has been stuck in a rut of trying to find one and then arguing about it. Pascal, C++, Java, Python, Scratch ... take a number."

He likens Python to a "package tour: safe, comfortable, blandly conventional. Choosing Python is the modern equivalent of the old adage, 'nobody ever got fired for buying IBM.'"

Guzdial is also not sure how long Python will be used as the main introductory programming language in academia. "I think Python has hit its tipping point, which may mean we have a couple more years before people say 'Python, what?'" ▣

---

**Further Reading**

Guo, Philip J.
**Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education, Google, Inc.**
http://bit.ly/1zB7ugb

Guttag, J.V.
**Introduction to Computation and Programming in Python, MIT Press (2013)**

Guzdial, Mark
**Exploring hypotheses about media computation.** *Proceedings of the ninth international ACM conference on international computing education research*
http://bit.ly/13SWMod

Enbody, R.J., Punch, W.F., and McCullen, M.,
**Python CS1 as preparation for C++ CS2.** *Proceedings of the 40th ACM technical symposium on Computer Science Education*
http://bit.ly/1tJjLu7

Pritchard, D. and Vasiga, T.
**CS Circles: An In-Browser Python Course for Beginners.** *Proceedings of the ACM technical symposium on Computer Science Education*
http://bit.ly/1z7h8UV

**How to Think Like a Computer Scientist.** *Learning with Python: Interactive Edition 2.0*
http://bit.ly/1tJkknG

---

**Esther Shein** is a freelance technology and business writer based in the Boston area.

---

# ACM Europe Protests H2020 Cuts

ACM Europe Chairman Fabrizio Gagliardi recently contacted European leaders in opposition to proposed budget reductions to Horizon 2020, the European Union's seven-year, 80-billion-euro research funding program.

In January, European Commission President Jean-Claude Juncker unveiled legislation that would remove 2.7 billion euros over 5.5 years from Horizon 2020, the EC's main funding stream supporting research through the year 2020, to devote those funds to economic stimulus through the creation of a European Fund for Strategic Investment. That investment, according to Juncker, would help get the sluggish European economy moving and create new jobs.

The largest share of the cuts would be directed at the European Institute of Innovation and Technology, which aims to spur innovation and entrepreneurship across Europe by bringing together universities, research labs, and companies to form "dynamic cross-border partnerships."

In letters to Juncker, European Council President Donald Tusk, and European Parliament President Martin Shulz, Gagliardi pointed out "the future success of Europe requires Europe to consolidate and advance its position at the forefront of scientific innovation. This goal requires major investments in fundamental research, especially in such critical domains as computing science."

Gagliardi said ACM Europe recommends the European Commission authorities and the European Council "preserve, in the announced cuts to H2020, the support to fundamental research and especially in computing science, given their direct relevance for the focus on innovation of the Investment Plan."
—Lawrence M. Fisher