

MAKING PROGRAMMING ACCESSIBLE FOR ALL

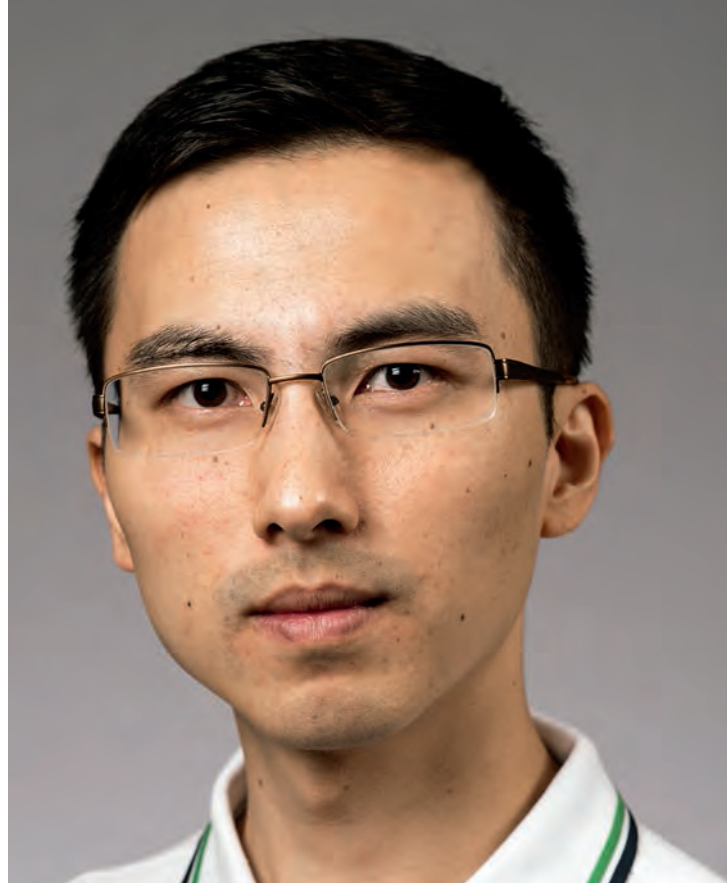
Computer science alum Philip Guo aims to lower the barriers to learning programming and data science.

By Eric Smalley | Connector Contributor

If Philip Guo were a superhero, his name might be “The Influencer.” Guo, SB ’05, MEng ’06, now teaches and conducts research at the University of California at San Diego, but his impact has been felt far beyond classrooms and labs. He has helped millions of people worldwide learn how to program, eased the fears and frustrations of thousands of doctoral students, and given thousands of people valuable insights into stereotypes and biases in the computing field.

Guo is an assistant professor in the UCSD Department of Cognitive Science, where he teaches human-computer interaction and conducts research on human factors, distance learning, and computing education. The path that led him there began with a childhood dream of studying computer science at MIT, and wound through an MEng thesis on tools for programmers, a doctoral thesis at Stanford on tools for data scientists, visiting researcher positions at Google, edX and Microsoft, and a postdoctoral position back at MIT’s Computer Science and Artificial Intelligence Laboratory (CSAIL).

Programming and data science are important for all college students, including those majoring in liberal arts, journalism, fine arts, and design, Guo says, adding: “That’s reflected in the job market as well.” However, many people shy away from the field because of its perceived difficulty, as well as stereotypes about programmers. “The popular image is of young people in hoodies crouched over a computer screen and being antisocial,” he says.



Philip Guo

Guo has dedicated himself to overturning these popular perceptions and making programming accessible to as many people as possible. One recent research paper tackles that misconception about programming only involving those hoodie-clad millennials. Instead, Guo studied adults aged 60 to 85 to uncover the cognitive and social challenges they face in learning how to program. The result was a proposal for a set of tools and techniques tailored to the needs of older adults.

The theme of Guo’s research is developing scalable ways to help people learn computer programming and data science. The centerpiece of this work is Python Tutor, a tool that allows people to write code in a browser and see automatically generated diagrams that illustrate what their code does. The tool has its roots in the software Guo developed for his MEng thesis that analyzes C and C++ code to let programmers see whether the code is running as expected. In addition to Python, the Python Tutor now works with Java, C, C++, Ruby, JavaScript, and TypeScript. “The C and C++ part of my visualizer tool actually uses a lot of the same ideas from my MEng thesis,” he says.

Python Tutor, which is free and open-source, is widely used in massive open online courses (MOOCs), traditional college courses and e-textbooks, Guo says. By his estimate, Python Tutor has, so far, been used by more than 3.5 million people in more than 180 countries to visualize more than 30 million lines of code.

In fact, Python Tutor has been the source of Guo's biggest impact, says Rob Miller, a professor of computer science who was Guo's postdoc advisor at MIT. The Web is full of tutorial sites and novice programming systems, but Python Tutor is unique because it offers a window inside the machine, automatically drawing pictures similar to those that a good instructor would draw on a blackboard, Miller says. Learning the skills of visualization and mental execution are critical to understanding how programs behave, he adds. "Every good programming teacher draws these kinds of pictures. Philip's is the first work I've seen that can create them automatically, for hundreds of simultaneous users, for every major language that people are trying to learn," Miller says. "That's a tremendous benefit to the world."


Beyond his core research and teaching, Guo has served as an informal mentor to thousands of PhD students, including many in fields far removed from computer science, by way of a virally popular e-book about his own experience earning a doctorate. The PhD Grind is a personal narrative of that six-year journey, which he wrote shortly after completing his degree. Guo says that, unlike numerous other works offering "how-to" advice for PhD students, his book allows people to identify with him. "I think it's a mirror neuron thing," he says. "People build empathy and find a way to commiserate."

Guo has also pointed a spotlight at biases in the computing field, particularly those that hinder female and minority students. He wrote a blog post, *Silent Technical Privilege* that detailed the advantage he gained from the stereotype of Asian males as skilled programmers, and contrasted his experience with those of fellow students who didn't fit that stereotype. NPR and Slate picked up the blog post, and he has since contributed to research on barriers confronting female programmers, co-authoring a paper about the challenges they face when they contribute to online forums.

Guo's current research is aimed at bringing the same types of tools he's developed for learning programming to the field of data science. Just as his programming visualization tool had its roots in his MEng thesis, this line of research builds on his PhD thesis, which helped researchers boost the productivity of their data analysis workflows. Guo's goals are to help people learn to work with multiple programming languages; develop tutorials to help people learn about data quality, numeracy, statistics, machine learning, and experimental design; and determine whether these types of tutorials can help novice data scientists avoid common experimenter biases, statistical misconceptions, and erroneous data interpretations. "The impact of this will be even bigger than programming because there are going to be many more people who do data analysis and data science than who are computer programmers," he says.

Whatever challenges Guo tackles in the future, his MIT experience has prepared him with more than just a thorough,

well-rounded computer science education. It also taught him the value of working with motivated and energizing faculty and students. "What MIT really brought to the table was providing a very intensive and passionate work environment," Guo says. "That has really long-lasting effects because, even years later, I'm able to have this determination and focus and work ethic that I and many of my peers developed during those years at MIT."

*See Python Tutor at pythontutor.com. Read the PhD Grind Blog at phdgrind.com and the *Silent Technical Privileges* blog at pgbovine.net/tech-privilege.htm. (Editor's Note: Philip Guo is unrelated to Margaret Guo, profiled elsewhere in this publication.)* 

The theme of Guo's research is developing scalable ways to help people learn computer programming and data science. The centerpiece of this work is Python Tutor, a tool that allows people to write code in a browser and see automatically generated diagrams that illustrate what their code does. The tool has its roots in the software Guo developed for his MEng thesis at MIT.