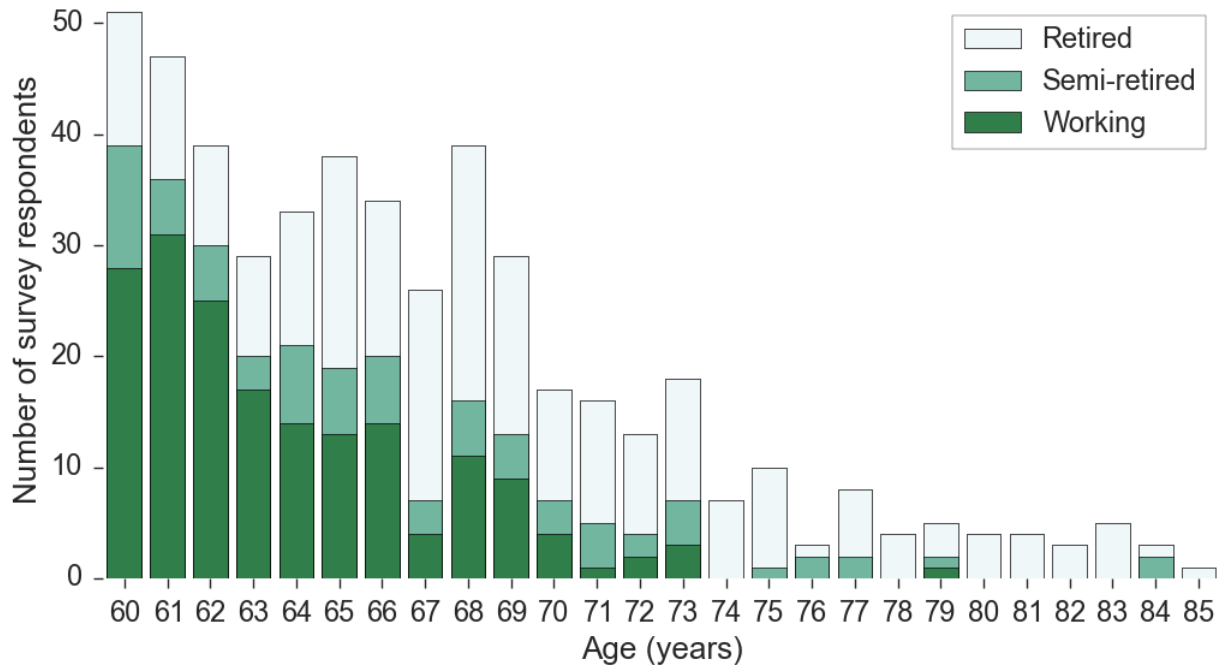


Geeking out in the golden years

April 19 2017



Philip Guo surveyed adults between the ages of 60 and 85 who were users of pythontutor.com and learning how to code. They were mix of retired, semi-retired and still working. Credit: Philip Guo, UC San Diego

Philip Guo caught the coding bug in high school, at a fairly typical age for a Millennial. Less typical is that the UC San Diego cognitive scientist is now eager to share his passion for programming with a different demographic. And it's not one you're thinking of - it's not elementary or middle school-aged kids. Guo wants to get adults age 60 and up.

In the first known [study](#) of older adults learning computer programming, Guo outlines his reasons: People are living and working longer. This is a growing segment of the population, and it's severely underserved by learn-to-code initiatives, which usually target college students and younger. Guo wants to change that. He would like this in-demand skill to become more broadly accessible.

"Computers are everywhere, and digital literacy is becoming more and more important," said Guo, assistant professor in the Department of Cognitive Science, who is also affiliated with UC San Diego's Design Lab and its Department of Computer Science and Engineering. "At one time, 1,000 years ago, most people didn't read or write - just some monks and select professionals could do it. I think in the future people will need to read and write in computer language as well. In the meantime, more could benefit from learning how to code."

Guo's study was recently awarded honorable mention by the world's leading organization in human-computer interaction, ACM SIGCHI. Guo will present his findings at the group's premier international conference, CHI, in May.

When prior human-computer interaction studies have focused on older adults at all, Guo said, it has been mostly as consumers of new technology, of [social networking sites](#) like Facebook, say, or ride-sharing services. While a few have investigated the creation of content, like blogging or making digital music, these have involved the use of existing apps. None, to his knowledge, have looked at older adults as makers of entirely new software applications, so he set out to learn about their motivations, their frustrations and if these provided clues to design opportunities.

The Study

For his study, Guo surveyed users of pythontutor.com. A web-based education tool that Guo started in 2010, Python Tutor helps those learning to program visualize their work. Step by step, it displays what a computer is doing with each line of code that it runs. More than 3.5 million people in more than 180 countries have now used Python Tutor, including those around the world taking MOOCs (massive open online courses). Despite its legacy name, the tool helps people supplement their studies not only of the Python programming language but also Java, JavaScript, Ruby, C and C++, all of which are commonly used to teach programming. The users of Python Tutor represent a wide range of demographic groups.

Guo's survey included 504 people between the ages of 60 and 85, from 52 different countries. Some were retired and semi-retired while others were still working.

What Guo discovered: Older adults are motivated to learn programming for a number of reasons. Some are age-related. They want to make up for missed opportunities during youth (22 percent) and keep their brains "challenged, fresh and sharp" as they age (19 percent). A few (5 percent) want to connect with younger family members.

Reasons not related to age include seeking continuing education for a current job (14 percent) and wanting to improve future job prospects (9 percent). A substantial group is in it just for personal enrichment: 19 percent to implement a specific hobby project idea, 15 percent for fun and entertainment, and 10 percent out of general interest.

Interestingly, 8 percent said they wanted to learn to teach others.

Python 2.7

```

1 def listSum(numbers):
2     if not numbers:
3         return 0
4     else:
5         (f, rest) = numbers
6         return f + listSum(rest)
7
8 myList = (1, (2, (3, None)))
9 total = listSum(myList)

```

[Edit code](#)

→ line that has just executed
→ next line to execute

Visualized using [Python Tutor](#) by [Philip Guo](#)

Frames

Global frame

- listSum
- myList

listSum

- numbers
- f 1
- rest

listSum

- numbers
- f 2
- rest

Objects

function
listSum(numbers)

tuple

0	1
1	

tuple

0	1
2	

tuple

0	1
3	None

The Python Tutor tool at work, showing code and diagrams. Philip Guo surveyed users of pythontutor.com for his study of older adults learning computer programming. Credit: Philip Guo, UC San Diego

Topping the list of frustrations for older students of coding was bad pedagogy. It was mentioned by 21 percent of the respondents and ranged from the use of jargon to sudden spikes in difficulty levels. Lack of real-world relevance came up 6 percent of the time. A 74-year-old retired physician wrote: "Most [tutorials] are offered by people who must know how to program but don't seem to have much training in teaching."

Other frustrations included a perceived decline in cognitive abilities (12 percent) and no human contact with tutors and peers (10 percent).

The study's limitations are tied in part to the instrument - self-reporting on an online survey - and in part to the survey respondents themselves. Most hailed from North America and other English-speaking nations. Most, 84 percent, identified themselves as male; this stat is consistent

with other surveys of online learning, especially in math and science topics. There was a diverse array of occupations reported, but the majority of those surveyed were STEM professionals, managers and technicians. These learners, Guo said, likely represent "early adopters" and "the more technology-literate and self-motivated end of the general population." He suggests future studies look both at in-person learning and at a broader swath of the public. But he expects the lessons learned from this group will generalize.

The Implications

Based on this first set of findings and using a learner-centered design approach, Guo proposes tailoring computer-programming tools and curricula specifically for older learners. He notes, for example, that many of his respondents seemed to take pride in their years and in their tech-savvy, so while it may be good to advertise products as targeting this age group, they should not appear patronizing. It might make sense to reframe lessons as brain-training games, like Lumosity, now popular among the older set.

Just as it's key to understand who the learners are so is understanding where they have trouble. Repetition and frequent examples might be good to implement, as well as more in-person courses or video-chat-based workshops, Guo said, which may lead to improvements in the teaching of programming not just for older adults but across the board.

Context matters, too. Lessons are more compelling when they are put into domains that people personally care about. And Guo recommends coding curricula that enable older adults to tell their life stories or family histories, for example, or write software that organizes health information or assists care-givers.

Guo, who is currently working on studies to extend coding education to

other underrepresented groups, advocates a computing future that is fully inclusive of all ages.

"There are a number of social implications when [older adults](#) have access to computer programming - not merely computer literacy," he said. "These range from providing engaging mental stimulation to greater gainful employment from the comfort of one's home."

By moving the tech industry away from its current focus on youth, Guo argues, we all stand to gain.

More information: [DOI: 10.1145/3025453.3025945](https://doi.org/10.1145/3025453.3025945)

Provided by University of California - San Diego

Citation: Geeking out in the golden years (2017, April 19) retrieved 18 April 2024 from <https://phys.org/news/2017-04-geeking-golden-years.html>

<p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p>
--