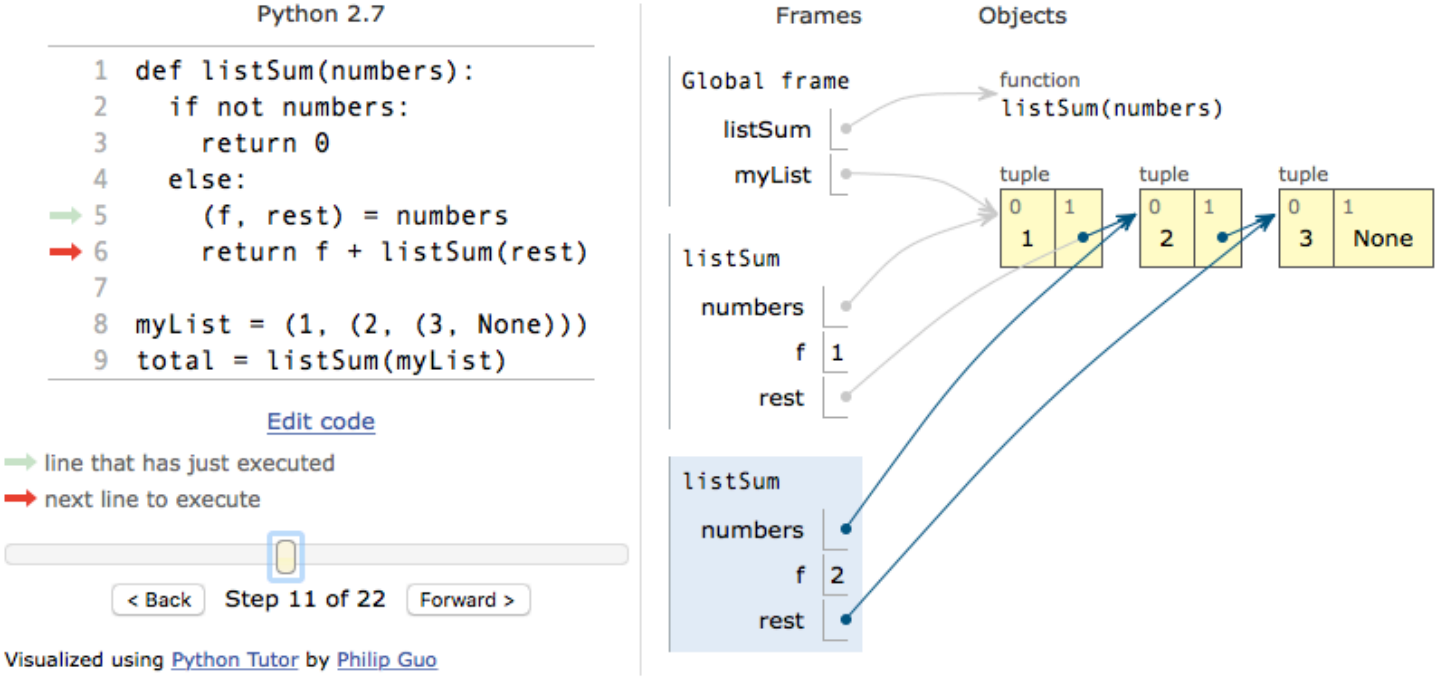
Python Tutor

Philip Guo (2013)

Visualizes the data structures and execution of programs. Runs on the web, is embeddable, and has achieved a degree of widespread use.



Python Tutor is a popular example of a multi-decade endeavor in computer science: program visualization system for pedagogic ends. It also overlaps with another effort called *software visualization*. Even the Atari 2600 had a BASIC cartridge exhibiting such characteristics, made by Warren Robinett, the creator of Adventure (2600) and Rocky's Boots (Apple II).

Some observations I pulled from surveys by Sorva and others (see references below):

• User motivation and engagement is critical. Sorva et al. (2013) argues that a constructionist orientation is desirable: learners are "makers who want to build things," which "can be harnessed for better learning."

• Level of abstraction of representation is an important choice. Are algorithms or program execution represented? Abstractions chosen reflect the aims of the system builders.

• Emphasis tends to be on generic representations. What if, instead, we allowed that special cased visual designs, perhaps by the programmer, were worthwhile?

For good surveys, see:

• Sorva, Juha. Visual Program Simulation in Introductory Programming Education. Aalto University, 2012.

• Sorva, Juha, Ville Karavirta, and Lauri Malmi. "A Review of Generic Program Visualization Systems for Introductory Programming Education." ACM Transactions on Computing Education (TOCE) 13, no. 4 (2013): 15.

See also:

• Guo, Philip J. "Online Python Tutor: Embeddable Web-Based Program Visualization for Cs Education." In Proceeding of the 44th ACM Technical Symposium on Computer Science Education, 579–584. ACM, 2013.