A Numpy-First Approach to Teaching CS1 to Natural Science Students

Elizabeth Patitsas
University of Toronto, Toronto, Canada
patitsas@cs.toronto.edu

ABSTRACT

Numpy (Numerical Python) and Scipy (Scientific Python) are Python libraries for doing numerical/scientific work that are popular with research scientists, as they allow for matrix-based computation in Python. I report on my initial experiences teaching a CS1 in Python to natural/social science students using a "numpy-first" approach. Students were taught about numpy arrays and matrix manipulations before learning lists and loops. I found this approach helped this audience appreciate the relevance of CS to their own fields, and possibly better learn topics such as logic and file I/O.

Categories and Subject Descriptors

K.3.2 [**Computers and Information Science Education**]: Pedagogy, non-majors

Keywords

CS1, non-majors, computer science education

1. INTRODUCTION

I will share my initial experience teaching a CS1 for non-majors using a "numpy-first" approach. The course is a 12-week, 100-level course directed at natural/social science students at a large research-intensive university, and covers the basics of Python with a focus on scientific applications. Half of the students are upper-level; most plan to go to grad school. Few students formally take any CS past this course.

The first time I taught the course (spring 2014), I followed the usual sequence of topics: variables and expressions; conditionals and logic; loops, strings and lists; file I/O; numpy/scipy. The students overwhelmingly perceived numpy/scipy as most relevant to their interests.

1.1 Motivation

Reflecting after the term, there were three issues I wanted to address for the next offering of the course:

1. Early in the term, students struggled to see the **immediate relevance of Python** to their lives, instead trusting it would be "useful later".
2. Students struggled with **conditionals and logic**, perhaps since we introduced both topics together.
3. **File I/O**: students had no notion of what a 'file' is from a CS perspective, nor what it meant to 'read' it.

1.2 Changes made

Teaching the class in spring 2015, I decided to move numpy earlier in the curriculum, so that the order was: variables and expressions; functions; numpy/scipy; conditionals and logic; loops, strings and lists; file I/O. My rationale for this was to address the three issues listed in subsection 1.1:

1. The early introduction of numpy/scipy makes the relevance of programming immediately clear to students. Students can promptly perform data analysis and graphing that is immediately usable in their other classes.
2. Numpy introducing array filtering, which means that students work with arrays of booleans before encountering conditionals. This reduces cognitive load.
3. Numpy provides the functions which encapsulate reading csv files to and from arrays. As a result, students can gain comfort with importing and exporting arrays before having to learn the details of `read` and `write`.

2. OBSERVATIONS ON NUMPY-FIRST

The numpy-first approach was well-received by students. It addressed the three issues in subsection 1.1. Students found the early focus on complex data analysis to be motivating for learning how to program.

This group of students has a background in linear algebra, and when they first see arrays/lists, they expect operators like + and * to add/multiple elements together, rather than concatenate/repeate. Introducing arrays before lists meant that the first time students saw a data structure its behaviour was congruent with their expectations.

The approach I took to teaching numpy involved teaching students about many of numpy/scipy's builtin functions. From early on, students got in the habit of looking up builtin functions rather than reinventing functions from scratch.

I needed to spend more time on numerical error and multiple return values than I expected. Both came up in using numpy for data analysis, and were confusing to students.

A downside of the numpy-first approach is that there are few resources for beginning programmers to learn about numpy: most numpy resources assume the audience can program already. There was also nothing in the class textbook on the topic. When numpy was at the end of term, students were better able to handle this.

Interestingly, when I introduced loops, many students expressed a distaste for them, preferring array manipulations.

Overall the numpy-first approach worked well for this science-focused audience, and I plan to use it again.

ITiCSE'15, July 6–8, 2015, Vilnius, Lithuania.
ACM 978-1-4503-3440-2/15/07.
http://dx.doi.org/10.1145/2729094.2754861 .