A first course in computer science is about a new way of solving problems: computationally. The choice of language can have an impact on how one approaches problems. In addition, the language can impact the kinds of problems that can be addressed in the course.

We have chosen Python as a the introductory language for CS1 students, majors and non-majors alike, based on our combined 30+ year experience in teaching undergraduate, introductory computer science at a Big Ten university. In that time, we have moved from Pascal, to C++ and now to Python.

We selected Python for the following reasons:

1. Python is a wonderful language for data analysis—allowing students to work on real problems analyzing data sets downloaded from a variety of sources on the Internet. For example, our students have analyzed breast cancer data, catalogued movie actor relationships, and predicted disruptions of satellites from solar storms.
2. Python is an interpreted language, allowing students to type in expressions and observe the results. In particular, most of the interpreters provide dynamic help to understand a function's or class's purpose and arguments.
3. Unlike many languages, Python intends to "keep it simple". In general, there are fewer options, many times only one option, for programmatic goals, reducing the cognitive load on a student learning a programming language. The focus of programming should be problem solving, and keeping the added load of learning the programming language itself should be minimized.
4. Python provides access to some advanced concepts, including advanced data structures (such as sets and dictionaries) and processing such as iteration through the elements in a complex data structure. Making these available as part of the basic Python package allows students to address real-world problems earlier in the studies.
5. Python contains features that are particularly helpful to novice programmers. Indention is required. A single equal sign is not allowed in a Boolean expression. All variables must be initialized.
6. Python contains all the typical programming constructs that one expects in a modern language, especially classes. We have found that transitioning to C++ to be reasonable for our students (see below).
7. Of particular importance, the Python community is very broad and provides access to a wide variety of packages for everything from science to art. Thus students can do meaningful work in their area of interest having learned the basics of Python.

---

1 The title has not been finalized.
8. Finally, after one semester of Python, students have sufficient command of the language to be able to write useful programs.

When we started our transition to Python, a CS1 text did not exist so we began writing one. The goals we have for the book are as follows:

1. The book is intended to teach the general concepts of CS1 to both majors and non-majors. Our goal is not to teach the language Python, but instead to teach CS1 concepts using Python as a vehicle. As such we cover general concepts such as hardware, complexity and software engineering as well as general concepts of programming using the Python language. While these topics are often considered “for majors”, general understanding of CS1 concepts will aid all introductory students.

2. We focus early on practical examples of programming, focusing on data manipulation as the general theme. All students will need to be able to manipulate data (sort it, apply statistics to it, graph it, convert it) no matter their area of expertise and we hope to exploit that need. As mentioned, we develop real-world examples that: observe trends in sunspot data (predicting the solar cycle), use statistics to classify breast-cancer data, combine results for searches to the IMDB (Internet Movie DataBase) and others. Such an approach is more gender neutral than other approaches, such as games or graphics, and more useful.

3. Our approach is an “object-use-first” approach where we use objects and their methods early while leaving the design and implementation of objects for later.

4. In the end, if a student takes no other programming course than one such as this, they can be productive programmers who understand general principles of computation and can produce useful, meaningful results in their respective fields of study.

Python is showing up in major courses, non-major courses, big schools, small schools, research institutions, first-year courses, and higher-level courses. Among that diverse group, our target is the CS1 course that includes a significant non-major component. We emphasize both problem-solving and programming skills. Non-majors should emerge with sufficient programming skills to perform computational analysis of data in their field. Majors should be prepared for a CS2 course in a different language. We expect our approach to be attractive for schools similar to ours, a large Big-10 university. However, by targeting CS1 courses with a mix of majors and non-majors we will not be eliminating smaller schools.

Such a CS1 course in Python could be helpful to the broader CS community by helping with recruitment and retention, especially with women. Python is a wonderful tool for manipulating data, both scientific and non-scientific. In particular, the availability of large data sets on the Internet provides problems in a variety of non-traditional areas. One can answer questions from cancer trends to ‘degrees of interaction’ among actors to when sunspots are next likely to disrupt communication satellites. Broadening the problems to solve while bringing in actual data promises to make the course attractive to a broader audience—especially those not interested in first-person-shooter games.
One of our goals with our Python CS1 course was to replace the existing C++ based course without changing the subsequent C++ based CS2 course. We conducted research on the performance of the first set of students in our CS2 course and have submitted the results to SIGCSE 2009. The Spring 2008 CS2 course (taught in C++) provided a unique opportunity to compare students who had CS1 in Python versus those with a different language. Using t-tests and multiple regression analysis we showed that students with a Python-based CS1 course were as well prepared for the C++ based CS2 course as those who had another language (C++ or Java). A draft can be found at http://www.cse.msu.edu/cgi-user/web/tech/document?ID=914