

Appendix: Yorktown High School Becomes a Python Workshop

by Lex Berezhny

Our transition to a Python workshop has proceeded rather smoothly, and allowed us to immediately start writing advanced code. We attribute this quick transition mainly to the interactive interpreter provided with Python. We have been able to quickly try out ideas, and get "live" feedback. This rapid development model has proven practical and effective in the creation of the two very exciting programs which are described below.

1. Student Portfolio System

The Student Portfolio System will allow students to track their academic progress through high school. By utilizing the Python language and the PostgreSQL database we were able to create a sophisticated system to maintain student data. The system enables students to select courses which meet their academic goals and graduation requirements. It also monitors their progress on important tests like SAT.

1.1 Description

Visually the portfolio system appears to its users in a folder format with tabs to each section. Students update/change certain data and get immediate feedback. The student portfolio system is intended to be practical and provides the maximum ease of use. One of its main features is assisting students in selecting classes depending on their prior course history. The program is organized into four main categories: course history, achievements (both academic and athletic), career plans, and standardized tests scores. The course history

section keeps a cumulative history of the classes a student has taken, and uses this data to help him/her select future classes. Academic and athletic achievements are very important to those planning to apply to colleges or universities. This section will help students keep growing records of their achievements. The career section will be the most helpful in planning long term goals for a student's educational path. They will be much better aware of what classes they need to take if they clearly state their career interests. Standardized tests such as PSAT, SAT I and II, and SOL are also very important. By organizing all of the scores in one place, it will make it easier for students to see which subject areas need work. As we continue developing the program these features will be improved upon and new ones added.

1.2 Technology

The front end is designed to implement a web-based interface. The interface is written with HTML 4 and CSS (Cascading Style Sheets), and uses the PNG graphics format with images. The engine is written in Python. This program has been developed using Python combined with a PostgreSQL database accessed through PyGreSQL. The HTML and most other interface aspects are generated using HTMLgen, which allows for use of existing HTML templates. This allows for greater independence between the HTML designer and the programmer. We plan to work on several versions of the program using different interface approaches and allow access from newer and older machines.

1.3 Perl vs. Python

Having written the first working version of this program in Perl, we found Python a great advantage over the Perl syntax. Python is much easier to read, maintain and understand. When several of us are working on the same project, we can spend less time explaining old code and more time writing new code. Writing advanced programs in Perl requires a veteran and with this being a student environment, many of us are writing our first large programs. Using Python has made all the difference.

1.4 Python Interpreter

The interactive interpreter that comes with Python makes testing code a dream. We can quickly test innovative ideas and see the results right away. This feature allows us to be more productive and to try better ways of doing things. By testing our code as we go along we write more concisely using functions and features that we wouldn't have used unless we were able to safely test them beforehand.

2. PyTicket

PyTicket is a program made to improve the efficiency of technical support within the school. A user simply fills out a form over the web, with problem description and physical location of the problem. Our technicians are then able to constantly check the system for any new tickets and respond within a reasonable amount of time. Using a systematic way of treating technical problems will be helpful in determining ways to prevent them in the future. Our technicians are advanced level students who are qualified to solve most computer-related (hardware or software) problems.

2.1 Description

PyTicket works similar to a bulletin board system in that a user can post a problem or a "ticket" and receive feedback on its status. It is an improvement on a bulletin board system, however, in that PyTicket ensures that everything is well organized and consistent. The PostgreSQL database is used to organize and assign "ticket" IDs, requestor contact information, and the physical location of the problem. Technicians are also assigned an id which is appended to each ticket whenever a technician decides to take the job. He or she is provided with a password to be able to update and edit each ticket after solving the problem.

2.2 Technology

The technology used by PyTicket is very similar to the Student Portfolio System. Python and PostgreSQL are again the main engines running the system.