Programming Leftovers

By *Roy Schestowitz*
Created *17/09/2021 - 3:28pm*
Submitted by Roy Schestowitz on Friday 17th of September 2021 03:28:03 PM Filed under Development [1]

- **Using functions more [2]**

  Bash functions seem to sit in a sweet spot between aliases and full blown scripts. I?ve defined a number of functions in my dotfiles which are all useful. Unlike aliases, they can take parameters and have greater scope for doing things; unlike scripts, they run in the context of the current shell which means, for example, that I can set a value in a variable during the course of a function?s execution and it?s available directly afterwards, in the same shell session.

- **Some notes on upgrading programs with Python's pip [3]**

  My primary use of Python's pip package manager is to install programs like the Python LSP server; I may install these into either a contained environment (a virtual environment or a PyPy one) or as a user package with 'pip install --user'. In either case, the day will come when there's a new version of the Python LSP server (or whatever) and I want to update to it. As I noted down back in my pip cheatsheet, the basic command I want here is 'pip install --upgrade <package>', possibly with '--user' as well. However, it turns out that there are some complexities and issues here, which ultimately come about because pip is not the same sort of package manager as Fedora's DNF or Debian's apt.

- **OpenBSD's pledge and unveil from Python [4]**

  Years ago, OpenBSD gained two new security system calls, pledge(2) (originally tame(2)) and unveil. In both, an application surrenders capabilities at run-time. The idea is to perform

initialization like usual, then drop capabilities before handling untrusted input, limiting unwanted side effects. This feature is applicable even where type safety isn?t an issue, such as Python, where a program might still get tricked into accessing sensitive files or making network connections when it shouldn?t. So how can a Python program access these system calls?

- **A good old-?fashioned Perl log analyzer [5]**

  A recent Lobsters post laud-ing the virtues of AWK remind-ed me that although the lan-guage is pow-er-ful and lightning-?fast, I usu-al-ly find myself exceed-ing its capa-bil-i-ties and reach-ing for Perl instead. One such appli-ca-tion is ana-lyz-ing volu-mi-nous log files such as the ones gen-er-at-ed by this blog. Yes, WordPress has stats, but I?ve nev-er let rein-ven-tion of the wheel get in the way of a good pro-gram-ming exercise.

- **Learning Path: Introduction to R [6]**

  Enhance your data science toolkit with our ?Introduction to R? learning path: from the basis of the syntax, to operations and functions, for solid programming foundations.

  R is one of the most popular programming, scripting, and markup languages. Written by statisticians for statisticians, it is an incredible tool for data exploration, data manipulation, visualization and data analysis. If you don?t have it yet in your pocket, or if you would like to build better foundations for your programming skills, this workshop series is what you were looking for.

Development

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] https://qmacro.org/autodidactics/2021/09/15/using-functions-more/
[3] https://utcc.utoronto.ca/~cks/space/blog/python/PipUpgradingPrograms
[4] https://nullprogram.com/blog/2021/09/15/
[5] https://phoenixtrap.com/2021/09/14/a-good-old-fashioned-perl-log-analyzer/
[6] https://www.r-bloggers.com/2021/09/learning-path-introduction-to-r/