Programming Leftovers

By Roy Schestowitz
Created 31/12/2020 - 8:57pm
Submitted by Roy Schestowitz on Thursday 31st of December 2020 08:57:02 PM Filed under Development [1]

- Pros and Cons of Building Your Own IoT Platform [2]

  Should I build my own IoT platform? Is it worth it to do it myself? The answer is, of course, it depends.

  You've got an idea for an awesome IoT project, and you're ready to get started making it into a reality. But there's a lot of options to choose from. Should you build your project from the ground up? Should you hire an IoT company to build it for you? Or should you take a middle path, using others' work as a foundation on which to build?


  This release of Is Parallel Programming Hard, And, If So, What Can You Do About It? features numerous improvements...

- Learn Lua by writing a "guess the number" game | Opensource.com [4]

  If you're a fan of scripting languages like Bash, Python, or Ruby, you might find Lua interesting. Lua is a dynamically typed, lightweight, efficient, and embeddable scripting language with an API to interface with C. It runs by interpreting bytecode with a register-based virtual machine, and it can be used for everything from procedural programming to functional programming to data-driven programming. It can even be used for object-oriented programming through the clever use of arrays, or tables, used to mimic classes.

  A great way to get a feel for a language is by writing a simple application you're already
familiar with. Recently, some Opensource.com correspondents have demonstrated how to use their favorite languages to create a number-guessing game. Lua is one of my favorites, so here's my Lua version of the guessing game.

- **Out of Control** [5]

  This leads us to ponder and not for either the first or the last time in the history of computer science the distinction between code and data. We stumble across this question even in the simplest cases. How would you describe the refactoring transformations above? Many would describe them in terms of separating the data from the code. Does that mean, then, that numerals.py contains data but not code? It's a valid Python module that initialises a variable to a list of string\-integer pairs. Sounds like code. Nothing says an essential qualification for something to be considered code is the presence of control flow.

  We use the word code freely, referring both to anything written in a programming language and, more specifically, to code (sic) whose primary concern is algorithm and operation rather than data structure and definition. Natural language is messy like that, filled with ambiguity, synecdoche and context dependency.

  If we want to be more rigorous, we could say that we have separated the code into code that abstracts operation and code that abstracts data. In other words, we are saying that Programs = Code and, given that Algorithms + Data Structures = Programs, therefore Algorithms + Data Structures = Code. This can be convenient and clear way to frame our thinking and describe what we have done. We also need to recognise, however, that it is just that: it is a thinking tool, a way of looking at things and reasoning about them rather than necessarily a comment on the intrinsic nature of those things; it is a tool for description, a way of rendering abstract concepts more concretely into conversation.

  If we confuse a point of view for the nature of things we will end up with a dichotomy that feels like Cartesian dualism. Just as Descartes claimed there were two distinct kinds of substance, physical and mental, we could end up claiming there are two distinct kinds of code code that is data and code that is operation.

  When we look to hardware, compilers or the foundations of computer science, such as Turing machines, we will not find clear support or a strict boundary for such separation. The indistinction runs deep. Although we have code and data segments in a process address, these enforce negotiable matters of convention and protection (e.g., the code or text segment is often read-only). Both code and data segments contain data, but the data in the code segment is intended to be understood through a filter of predefined expectations and an instruction set. On the other hand, it is also possible to treat data in the data segment as something to execute.

- **Perl weekly challenge 093** [6]

  Using a similar approach than James Curtis-Smith, the solution looks at points with equal slope to see if they are in a straight line. Being less literate in Raku, using classes help me to organize coding ideas.
This exercise gives me the opportunity to work with the type BagHash. The highest value of the slopes stored in a BagHash gives the number of points in a straight line. Happily, the first example in Raku documentation is for a Class Point, an example reused in this solution.

I'm Making Headway Now [7]

Last January there was a post on reddit which claimed that my module JSON::Parse was not only failing some of the JSON Test Suite tests, but also crashing on one of them. Anyway I should have got around to doing something about it sooner, but here are my conclusions.

First of all there was a crash on one of the files, which went something like this: [{"":[{"":{""}}, repeated about 100,000 times. Investigating it using a Linode, I found that after 80,000 open brackets the stack was overflowing, causing the crash to occur. If I added a printf in the midst of my code the printf would cause the stack overflow, so it wasn't actually due to my code but just because the stack size seems to be quite small on Linux.

There are various things one could do to tackle this, but it does seem a bit unlikely that anyone would want to have that many open brackets, so what I did as a strategy was to add a "max_depth" of parsing after which it would stop. I thought 10,000 open { and ] would be enough for anyone, and it would satisfy the people who want to run the JSON Test Suite tests, but I also added an option for the user to alter the max depth and get the max depth as well.

Jean-François Fortin Tam: Blogging about Python desktop apps improvements on Planet Python[8]

Hi, fellow pythonistas! Before I start publishing future Python-related posts to this aggregator, I would like to shortly introduce myself and the reason for this blog's presence on the planet.

[...]

Here I blog mainly about new releases and improvements in my Python software apps (which means GTG lately, but I also have a couple of pythonic utility apps I've been meaning to publish sometime soon), and sometimes write about performance optimization in software applications in general, or how a particular bug was solved. As such, my blog posts tend to be ?applied Python? type of content rather than theoretical tutorial-style blog posts.

10 examples of using Python in 2020 | Opensource.com [9]

Each year, Opensource.com publishes various articles about Python to pique new users' interest and help long-time Pythonistas expand their skills. The following are Opensource.com's top 10 articles about Python in 2020.
MySQL is an RDBMS (Relational Database Management System) which is owned by the Oracle Corporation and inherited from the standard SQL. It allows access and manipulation of Databases. Whoever knows the word Database? must have knowledge of Primary and Foreign keys. There is no concept of a relational database without the existence and idea of the concepts of Primary Keys and Foreign Keys. So in this article, we are going to learn about the importance and correct use of Primary and Foreign keys in MySQL.

The primary key can be any field or column of a table, which should be a unique and non-null value for each record or a row.

The Foreign key is a field that contains the primary key of some other table to establish a connection between each other.

Let's have a look at the syntax and different examples to create primary and foreign keys in MySQL.

MySQL provides a lot of commands, which are needed while managing a database. For example, we often need to get some data from different tables based on some condition. MySQL then provides different types of joins to get the desired results. Let's learn LEFT JOIN AND RIGHT JOIN of MySQL.

There is no such statement as FULL OUTER JOIN in SQL, but we can use a simple JOIN to get the same results or by simply using a SELECT statement over two different tables.

Otherwise, MySQL provides LEFT JOIN and RIGHT JOIN to get the records or rows from the right or left table, respectively. Let's try a couple of different examples to get the desired results using appropriate joins.

Source URL: http://www.tuxmachines.org/node/146018

Links:
[9] https://opensource.com/article/20/12/python