Published on *Tux Machines* (http://www.tuxmachines.org)

Python Programming

By *Roy Schestowitz*
Created *06/04/2020 - 12:35pm*
Submitted by Roy Schestowitz on Monday 6th of April 2020 12:35:36 PM Filed under Development [1]

- **Beets ? music tagger and library organizer using the MusicBrainz database [2]**

  > The music scene is where I?m happiest. As an amateur musician, I devote an inordinate amount of time developing my technique, practicing, practicing, and practicing. I also love listening to music, both live and recorded. Linux is my other passion. It?s endowed with a bewildering arsenal of open source multimedia software, so I?ve invested a lot of time reviewing a fair chunk.
  >
  > Over the years I?ve amassed a bountiful eclectic music collection. In my formative years, that was mostly pop music but over the years I?ve progressed to classical, jazz, blues, techno, and even a smattering of heavy metal. While I have a large collection of vinyl and CDs, I mostly listen to FLAC files these days. FLAC is an audio coding format for lossless compression of digital audio.
  >
  > Where does beets step in? If your music collection is in a sorry state of affairs with missing or incomplete track details, metadata, duplicate tracks, missing tracks, then beets might just fit the bill. Besides metadata, the software also grabs album art, lyrics, transcodes audio to a wide variety of formats, and much more. It?s a library that?s designed to be as flexible as possible.

- **Python Lists [3]**

  > Python includes a number of sequential data types that allow you to store collections of data in an organized and efficient way. The basic sequence types are lists, tuples, and range objects.
  >
  > This article goes through the Python lists. We?ll show you how to create a list, slice and sort a list, add or remove elements from a list, and so on.
  >
  > Lists are mutable sequences, which means that they can be changed after creation. Lists are

one of the most commonly used data types in Python and are generally used to store collections of items of the same type.

- **Flask Delicious Tutorial : Building a Library Management System Part 1 - Planningc [4]**

  This tutorial aims at helping all learners of Python: businessmen, students, tinkerer and teachers learn web development with Python using Flask. One of the joys of Python is fun programming and web development seems to bring another level of happiness. This is dedicated once again to all Python learners! I've pulled in this tutorial from my own experience teaching Python and client requests. So be sure to roll up your sleeves as it'll be more than a toy app and requires some work as real world apps have more features. I'd be not so nice if in real life you get to develop something without a project statement. I'll also cover some secret techniques i found along my Python dev experience!

- **Effective Developers Leverage Their Toolset [5]**

  Last week I did a couple of shared screen sessions debugging and teaching.

  I paused and reflected on the tools I used and how I sharpened my sword over the years.

  This is not an article on how to deploy software with Docker, how to use git, or how to set up your env, although it has some shell and Vim goodness.

  It's more about how small tweaks made me more productive as a programmer and learner.

- **PyDev of the Week: Pablo Galindo Salgado [6]**

  This week we welcome Pablo Galindo Salgado (@pyblogsal) as our PyDev of the Week! Pablo is a core developer of the Python programming language. He is also a speaker at several Python related conferences.

  [...]

  I am currently working at Bloomberg L.P. in the Python infrastructure team, supporting all our Python developers and providing critical infrastructure and libraries to make sure everyone has better experience programming in Python. But before working on the Software industry I used to be in academia as a theoretical physicist researching general relativity and in particular, black hole physics. This is something that I still do as a hobby (although without the pressures of publication and funding) because I still love it! For instance, I have given some talks in some Python conferences related to this (https://www.youtube.com/watch?v=p0Fc2jWVbrk) and I continue developing and researching improved algorithms to simulate and visualize different spacetimes. For example, here you have some simulated Kerr Newman black holes with accretion disks around them I have worked on recently:

- **Learn Python Identity Operator and Difference Between ?==? and ?IS? Operator [7]**

  This article is mainly curated to explain an important operator in python (?IDENTITY OPERATOR?) and how an identity operator differs (is, is not) from comparison operator (==).

- **What is Celery beat and how to use it ? part 2, patterns and caveats [8]**

  Celery beat is a nice Celery?s add-on for automatic scheduling periodic tasks (e.g. every hour). For more basic information, see part 1 ? What is Celery beat and how to use it.

  In this part, we?re gonna talk about common applications of Celery beat, reoccurring patterns and pitfalls waiting for you.

Development

Source URL: http://www.tuxmachines.org/node/136061

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] https://www.linuxlinks.com/beets-music-tagger-library-organizer-musicbrainz-database/
[3] https://linuxize.com/post/python-list/
[4] https://www.codementor.io/@abdurrahmaanj/flask-delicious-tutorial-building-a-library-management-system-part-1-planning-158qfbl7kc
[5] https://pybit.es/developer-tools.html
[6] http://www.blog.pythonlibrary.org/2020/04/06/pydev-of-the-week-pablo-galindo-salgado/
[7] https://www.tecmint.com/learn-python-identity-operator/
[8] https://breadcrumbscollector.tech/what-is-celery-beat-and-how-to-use-it-part-2-patterns-and-caveats/