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

Programming/Development: Zato, Wing, Receiving Code Review, CoffeeScript and BASIC

By *Roy Schestowitz*
Created *18/06/2019 - 3:24am*
Submitted by Roy Schestowitz on Tuesday 18th of June 2019 03:24:22 AM Filed under Development [1]

- **Zato 3.1 Released - Open-Source Python-based API Integrations and Backend Application Server [2]**

  > The newest version of Zato, the open-source Python-based enterprise API integrations platform and backend application server, is out with a lot of interesting features, changes and additions.

- **Extending Wing with Python (Part Two) [3]**

  > To debug extension scripts written for Wing, you will need to set up a new project that is configured so that Wing can debug itself. The manual steps for doing this are documented in Debugging Extension Scripts. However, let's use an extension script to do this automatically.

- **Robbie Harwood: Receiving Code Review [4]**

  > From a maintainer's perspective, that's the primary role of code review: to ensure project quality and continued maintainability. But there's an important secondary purpose as well: to help contributors (and potential contributors) learn and grow. In other words, receiving code review is a learning and growth opportunity, and should be approached as such.
  >
  > And so, first and foremost: code review is not a judgement on you. It's a second set of eyes, and both of you are trying to make sure the changes are good. If they didn't want the change in the project, they'd say so! Subtlety isn't what's happening here. And besides, if anyone were perfect, we would do code review.

> Which leads into: everyone needs code review. No change is too small for it, and no one is perfect. I've broken builds by changing only documentation, and flagged potential security issues from developers who have been coding almost as long as I've been alive. (And they've done the same to me!) That's normal. That's life. That's code review.
>
> And it's fine, because we don't need it to be perfect on the first try. Contributing to open source isn't a school exam where we get a single attempt and it's most of the grade. We're concerned only with improving our software, and if there's grading at all, it's externally imposed (e.g., by an employer).

- **Best Free Books to Learn about CoffeeScript [5]**

  > CoffeeScript is a very succinct programming language that transcompiles into JavaScript, so there is no interpretation at runtime. The syntax is inspired by Ruby, Python and Haskell, and implements many features from these three languages.
  >
  > CoffeeScript is closely related to JavaScript without having its eccentricities. However, CoffeeScript offers more than fixing many of the oddities of JavaScript, as it has some useful features including array comprehensions, prototype aliases and classes. It allows developers to write less code to get more done.

- **10 PRINT Memorial in New Hampshire marks the birthplace of BASIC [6]**

  > After just over 55 years, the birthplace of BASIC has been honoured with a memorial marker in New Hampshire, USA.
  >
  > Thanks to a campaign by local paper columnist David Brooks, the New Hampshire Historical Highway Marker was installed earlier this month.
  >
  > Professor John Kemeny, Maths professor Thomas Kurtz, and a group undergraduate students at Dartmouth College (pics) created BASIC (Beginner's All-purpose Symbolic Instruction Code). The first program ran on 1 May 1964.

Development

Source URL: http://www.tuxmachines.org/node/124984

Links:
[1] http://www.tuxmachines.org/taxonomy/term/145
[2] https://zato.io/blog/posts/zato-3.1-released.html
[3] https://wingware.com/blog/scripting-2
[4] https://mivehind.net/2019/06/17/receiving-code-review/
[5] https://www.linuxlinks.com/best-free-books-learn-coffeescript/

[6] https://www.theregister.co.uk/2019/06/17/plaque_for_basic_birthplace_new_hampshire/