

Programming Leftovers

By *Roy Schestowitz*

Created *13/12/2019 - 9:31am*

Submitted by Roy Schestowitz on Friday 13th of December 2019 09:31:32 AM Filed under [Development](#) [1]

- [Fedora 32 Will Feature Bleeding-Edge Compilers Again With LLVM 10 + GCC 10](#) [2]

Fedora Linux is on track to deliver another bleeding-edge compiler toolchain stack with Fedora 32 due out this spring.

Fedora's spring releases have tended to always introduce new GNU Compiler Collection (GCC) releases that are generally out a few weeks before the April~May Fedora releases. Thanks to Red Hat employing several GCC developers that collaborate with Fedora, they tend to stick to ensuring Fedora ships new GCC releases quite quickly while managing minimal bugs -- in part due to tracking GCC development snapshots well before launch to begin the package rebuilds.

- [What makes Python a great language?](#) [3]

I know I'm far from the only person who has opined about this topic, but figured I'd take my turn.

A while ago I hinted on Twitter that I have Thoughts(tm) about the future of Python, and while this is not going to be that post, this is going to be important background for when I do share those thoughts.

If you came expecting a well researched article full of citations to peer-reviewed literature, you came to the wrong place. Similarly if you were hoping for unbiased and objective analysis. I'm not even going to link to external sources for definitions. This is literally just me on a soap box, and you can take it or leave it.

I'm also deliberately not talking about CPython the runtime, pip the package manager, venv the %PATH% manipulator, or PyPI the ecosystem. This post is about the Python language.

My hope is that you will get some ideas for thinking about why some programming languages feel better than others, even if you don't agree that Python feels better than most.

-

[Python String Replace](#) [4]

In this article, we will talk about how to replace a substring inside a string in Python, using the `replace()` method. `replace()` Method In Python, strings are represented as immutable `str` objects. The `str` class comes with many methods that allow you to manipulate strings. The `replace()` method takes the following syntax: `str.replace(old, new[, maxreplace])` `str` - The string you are working with. `old` ? The substring you want to replace.

[Development](#)

Source URL: <http://www.tuxmachines.org/node/131663>

Links:

[1] <http://www.tuxmachines.org/taxonomy/term/145>

[2] https://www.phoronix.com/scan.php?page=news_item&px=Fedora-32-GCC-10-LLVM-10

[3] <http://stevedower.id.au/blog/python-a-great-language/>

[4] <https://linuxize.com/post/replace-string-python/>