

Python, Randomization, E2E – it's all about Random and some good reasons to learn and leverage on Python

Order is a very relative notion, it is specific for each of us. "The world is my representation" as Arthur "Chop" Schopenhauer says. At the same time, experience teaches us "Every cloud has a silver lining" so modestly I discovered that a relative "disorder" is often more efficient than an absolute "order". Right? Ooh slow down a bit! It is not an easy-peasy Q/A!

For sure, you can ask yourself why such interrogations are part of a so called post dedicated to Python. Well, first of all, it make sense for me for several reasons:

Never bad to introduce some philosophy in programming practice. Personally, it reconciles two of my favorite hobbies (bad programming & cheap philosophy) to one : headache! More seriously, these thoughts come from a practical and real-world experience.

Let me tell you the story: after a long, long aging process, as a PO, I achieved to build a robust testing strategy with CodeceptJS for a backoffice! But then I discovered that the e2e sequential execution of the suite created a bias (let's called them errors). OMG! Then I was wondering how to introduce randomization inside the testing suite execution in order to reduce this bias to the maximum. The purpose is to avoid the false positive nightmare!

Meanwhile, in a completely chaotic, confused, disordered, disorganized, topsy-turvy, helter-skelter, pell-mell, upside-down, higgledy-piggledy, hugger-mugger, harum-scarum, snafu, slipshod, unplanned, erratic, strayed, hit-and-miss, incidental, spontaneous, uncoordinated, willy-nilly, devil-may-care, unpremeditated, reckless, cockeyed, hit-and-miss, fluky, incidental... anarchic way, I am learning advanced Python techniques (NLP, facial recognition ...) but also fundamentals like the difference between a "tuple" and a "dictionary". So, it was logic that Python popped up in my mind to bring this randomness, given the complexity of doing the same thing in Bash, a language that I am far from mastering!

Incidentally, this experiment proves me that Python is:

Easy to read: Python is easy to read and most of the language makes sense at a glimpse. This makes finding issues a lot easier than more complicated languages.

Portability: Python runs on many platforms and systems, meaning your programs can reach a wider audience. It can easily replace Bash or Ruby of instance.

Two qualities that gave Python a serious advantage in what I am looking for: to create on-demand order or disorder 😊

Order: when it comes to browsing, summarizing and indexing hundreds of texts in order to store them in a database itself MySQL and/or NoSQL.

Disorder: when it comes to run tests in parallel, for example, and thereby overcome a bias linked to a sequential execution.

Oddly, sometimes, Disorder proves to be more efficient and effective than Order.

So, this post presents my quick researches on how-to handle random in Python. Having in mind that the final objective is to load randomly testing files that achieve a complete Backoffice's assessment without increasing errors into the process.

By the way, it reveals a third and fourth Python qualities:

Increasing productivity: how much work you can accomplish in a given time with few lines of code.

Relieving Boring Stuff in Bullshit Jobs: Thanks to Al Sweigart and David Graeber to have enabled me to create this "tuna-mayo-sandwich" concept.

So, now, let's move on and show some code...