Subject: Re: Beginner, Question
Posted by Jeremy Bailin on Thu, 03 Sep 2015 17:28:15 GMT
View Forum Message <> Reply to Message

On Wednesday, September 2, 2015 at 3:22:04 PM UTC-5, Paulo Penteado wrote:
> Also not trying to start a war, I can add my view on this. From the perspective of an astronomer who has worked a lot with IDL and Python (disclaimer: I have a lot more experience and knowledge in IDL than in Python). My personal reasons for preferring IDL over Python as a language are, most importantly:
>
> 1) Better vectorization. Numpy is very advanced, but because it is just a module, vectorization is not (for now, at least) part of Python's semantics. This makes vector operations (particularly in arrays that are not 1D) awkward in Python, requiring many transformations between Numpy and native objects, and/or lots of loops.
>
> 2) Portability is way better in IDL. Getting some complex program to work on Python, due to their many dependencies on specific versions of other libraries, is often a lot of (platform-specific) trouble. This is what causes several vendors/distributors to package their whole thing into a self-contained Python distribution (Anaconda, Ureka, yt, etc). Which results in one having several independent Python distributions installed, each one usable for one software package, and is a big obstacle to write code that uses functionality from more than one of those distributions. Additionally, code that works now may easily get broken by the next non-backward compatible update in some library. Plus, there is a whole other source of trouble with the Python 2 vs Python 3 issue, which has existed for many years, and probably will keep being an issue for a long time.
>
> 3) Better visualization libraries in IDL, which are built-in, unlike Python, which may need messing around with installing several other platform-dependent libraries or getting them to work with the kind of graphics device in use.
>
> 4) More functional IDE and debugger in IDL. I have been through many Python IDEs, and I was never satisfied. None seemed to work for debugging as well as IDL's. In Python IDEs there is always some trouble setting the environment or the program execution, or things that seem to decide not to work.
>
> I do recognize that Python has some advantages. Mostly, being free and the greater number general-purpose (as opposed to scientific) modules available. That said, I have to note that languages are not chosen just in an abstract way, considering their intrinsic characteristics. Each project has different constraints, such as availability of useful libraries, existing code, availability of IDL licenses, and language preferences of other people involved in the project or future code users. All of these influence what I find to be the better choice for that particular project.
>
> Paulo

Yes, all good points.

I'm never sure whether speed issues are intrinsic to the languages/libraries or to my use of them. I know how to write fast IDL code -- I leap straight to the histogram/value_locate/etc tricks immediately. I don't know the equivalents in Python, i.e. what things are fast, what things are slow, what tricks to use. So usually my IDL is faster than my Python. On the other hand, I know Python people who say the exact opposite -- their Python is usually faster than their IDL. I'm not sure how my IDL compares to their Python in speed, though!

-Jeremy.