Vincent Comaroto

Python is a language that I have in the past and now still have a love hate relationship with. I will be the first to admit, my view of python is incredibly biased, as 90-95% of my daily programming is done in Java, with maybe a bit of C strewn in. With this said, going from such a “verbose” language like java, to a language like python where it allows a lot of assumption is quite jarring.

I'll start out with the things I enjoy about python and then later move on to the downsides from my personal taste. Python, I have found, is incredibly nice for making a quick and nice program with a lot of modularity or imported libraries. This is very nice for time when I am trying to learn a new system (“machine learning. video tracking,data processing,etc”). These are things that in older languages like java or c can be quite the hassle to get working and even then may not be as efficient or clean as they would be in python. Another thing I do like when I remember to get used to it is the speed that coding can be done. Due to the dynamic typing, lack of semicolons etc, once I get into the groove of python, I can write a program much faster and without as much fluff as I would need in other languages,this however is a double edged sword as I will go into in the parts I don't like about python which are all somewhat related to the things I like.

On to the negatives of Python, again coming from my overtly biased perspective. I'll start with the new things, firstly setup. It always seems that no matter how smooth or how nice an IDE I use for python be it Pycharm, Jupyter, Sublime, etc, they always end up being a much bigger pain to get running to to having to select the right interpreters, different modules being ran for each one etc. On a similar vein, debugging in python I have found can be quite annoying when it's just downright difficult. I should say now I really dislike pythons mandatory indentation rules. While I understand that if they can become second nature it is fast, simple, and can appear to be very clean code but for me this has never been the case. I have found that the lack of anyclear parenthesis or semicolons or other similar symbols in python code makes it much harder for me to grasp scope of the program at a glance and I find myself having to put those things in in my head to fully grasp the program. This is something I'm sure can be unlearned with time, but I don't know If I'd ever truly get used to it. This is also quite subjective but I've never gotten the feeling from python that it is meant to be very scalable.Due to it not being as Object orientated it doesn't feel to me like it would be as useful in large scale projects as opposed to one file programs. This is compounded by the quick but hard to quickly read dynamic typing of python, In which it can make it hard to know what type of data a variable is without knowing the larger context of its use. So to end my rant, python is great for the user to write, but not so much to later analyze quickly.

Some resources I used and found helpful were the python docs which were very user friendly and of course assorted youtube videos and stack overflow. Comparing the python docs to other languages and programs I have used was a breath of fresh air considering how nice and organized it was. They were broken down into neat and concise explanations as well as detailed examples on how to implement things. This proved very important when learning to use the built in python CSV read which otherwise would have had me lost. The other side was for more targeted issues about when some random issue I would have just wouldn't make sense, I would resort to googling the issue and watching a youtube video or reading a forum post about it. While these have a much larger variation of quality, they can prove to be a lot more targeted to the current issue at hand.