The Overhyped Language: Python

Python is an undeniably powerful and most versatile language that has stormed the field of programming. It is widely used for web Development, Data Analysis, Machine learning, Artificial Intelligence and More. But It has various problems as well. In this article, we will look at the pros and cons of python, and we will try to explain why python is not always the best choice for programmers

Hype Around Python

Python is often lauded for usability and readability. Python's extensive standard library and multitude of open-source packages have made it a popular choice for quick prototyping and developing complex applications with ease.

Furthermore, python’s rise can also be attributed to its critical technologies like AI, Machine learning and Data Science. For instance, popular libraries like Pandas are all based on Python.

But there are some(or a lot of) problems with Python, and I will try to talk about them one by one:

Readability:

Although its easy to learn and read, this is valid only at the fundamentals of python. Once you get past the “hello World” program, it can really get ugly and counter intuitive.

Multiple ways for a single task:

The pythonic philosophy –

“There should be one -- -- and preferably only one – obvious way to do it .”

As someone who loves to work within rigid framework and rules, I love this philosophy! As someone who writes Python, I wish Python actually stuck to this philosophy.

For example –

400 ways (more or less) to interpolate strings –

user = {'name': "Robin"}

print(f"Hello {user['name']}!")

print("Hello {name}!".format(\*\*user))

print("Hello %(name)s!" % user)

If there was a unique and obvious use case for each of these than that would be a thing, but obviously its not.

Then there are 69 top level functions to memorize. GvR’s explanation sounds great, but in reality it makes things confusing.

Forced Indentation:

Absence of parenthesis is really weird thing in python. What kind of efforts are required to put two parenthesis? Some love the indentation as it enforces consistency and a degree of readability. Some hate it because it enforces wrong consistency. Its quite subjective, but I still hate that Indentation.

A Fragmented Language:

For a very long time, Python had two major versions running around in the wild. They are, of course, Python 2 and 3, and they’re incompatible with each other. So you were put in this weird situation where a MAANG company would say “come on guys, learn some Python, it’s cool!” and then you would go to download it, and you would be given the choice between versions 2 and 3, and maybe MAANG would tell you “pick 3!” and others would tell you “pick 2!” and then you would find out that 3 barely had any compatible libraries and no one used it, and the only reason why 3 was developed was for adding parentheses to the print function and emoji support. Only recently has the situation gotten any better. Ubuntu, for instance, recently phased out Python 2.

Map and Lists –

Map doesn't return a list, even though the whole point of a mapping function is to create one list from another. Instead it returns a map object, which is pretty much useless since it's missing append, reverse, etc. So, you always have to wrap it in list(), or use a list comprehension, which, speaking of...

List comprehensions are held up as an excellent recent-ish addition to Python. People say they're readable. That's true for simple examples (e.g. [x\*\*2 for x in range(10)]) but horribly untrue for slightly more complex examples (e.g. [[row[i] for row in matrix] for i in range(4)]). I chalk this up to...

Weird ordering in ternary/one-line expressions. Most languages follow a consistent order where first you declare conditions, then you do stuff based the on those conditions:

if user.isSignedIn then user.greet else error

for user in signedInUsers do user.greet

Python does this in the opposite order:

user.greet if user.isSignedIn else error

[user.greet for user in signedInUsers]

This is fine for simple examples. It's bad for more complex logic because you have to first find the middle of the expression before you can really understand what you're reading.

Problems with Culture:

Some people will argue saying “You don’t understand Python” and thereby not acknowledge criticisms of their favorite language. These Pythonists will always try to defend their language. They have a bit of superiority complex as well.

In Conclusion, Python is an amazing, undeniably versatile language. However, we need to understand that no language is fit for all use in the universe. While python may excel in certain domains, it may not be useful in high performing tasks.