Python : Background to Present-Day

Monika Baloda

History of Python and Python3

The beginning of the language can be traced back to 1989, with successive updates over time, we are now in a world of Python-3. Python-3 was released in 2008, it is a major improvement over its previous version. Some of the key features that distinguish it from the earlier versions are:

1. Unicode support: This means that Python 3 can handle text in any language, including non-Latin scripts, more easily.
2. Print function: In Python 3's print function is more consistent with other built-in functions and allows for more flexible and powerful output formatting.
3. Division: In Python 3, the division operator (/) always returns a float, unlike Python 2 which returns an integer if both operands are integers. This can lead to unexpected results and requires extra code to handle correctly.
4. Type hints: Python 3 includes support for type hints, allowing developers to specify the expected type of function arguments and return values. This can help catch errors earlier and make code easier to read and maintain.
5. Asynchronous programming: Python 3 includes built-in support for asynchronous programming, making it easier to write code that can handle multiple tasks simultaneously without blocking.

Comparing Python to Other Languages:

According to an article in Hacker news, python became the most popular language in the United States in 2014. Since then, it enjoys the tag of most popular language. In the lecture, we compare it with languages. Let's start with an example, we need to store a value 2 in a variable then we write it as follows :

In Python : a=2

In Java or C++: int a=2

In python we need not to declare data-type, it automatically recognizes or assigns the best one. Sometimes we just want that so Python is our popular choice. However, we are expert software developers and want greater control over data types so we may choose C++ over Python. But since data scientists don't care about it much, Python is their choice.

While all programming languages are developed for a particular set of tasks, we cannot say that one language is universally the best. For example, for object oriented programming, Java is considered one of the best. The C++ and now C# enjoys the expertise in the tasks where we want basics to be precise in programming. But when the programmer needs to use some functions often then it makes sense to write those functions already in the library so that it becomes easy for her to directly use and devote more time to the actual analysis.

Python vs R for data science:

One of the most asked questions by an aspiring data scientist or analyst is which language she should learn : Python or R? The general answer which I also got from the seniors is that learn any one of them and become an expert in one of the languages instead of sailing on two boats together. The datacamp blog referred to in the notes suggests that both languages can be used as a complement to each other in the long run.

If I'm asked to give two reasons why to choose python, then I will say that first it is high readability and interpretability. The second reason comes from the saying, as they say that people make the places great, the community of users of python makes it a great language. It is easy to use libraries designed for all kinds of purposes.

However this doesn't mean that python is superior to R. R language is designed by statisticians for technically sound statistical computations. The R also enjoys huge community backed support sitting in the top statistics departments in the world. However tasks like web design cannot be done by R while one can do it in python.

Overall, we conclude that both languages can be complementary rather than mutually exclusive to each other. The key difference between the two can be found here and here.