Write a short essay discussing the heritage and philosophy, development platforms, characteristics, and strengths & weaknesses of the new programming language.

SDV602 Milestone Three

New Language Reflection Essay

19/11/23

Jayden Houghton – 13524050

# Introduction

This short essay will discuss the Python programming language. It will demonstrate my understanding of Python, as well as identify the purpose and characteristics of it as a language.

# Heritage and Philosophy

Python was first developed in the late 1980s by Guido van Rossum, with its first release in 1991. It has had many updates since then, including the major versions 2.0 being released in 2000 and 3.0 being released in 2008 (Wikipedia contributors, Retrieved 2023). The main reason for why Python was developed, was to create a language that emphasised code readability, and allowed programmers to express concepts in fewer lines of code (History of Python, 2023).

Python was originally started as a hobby project, with Van Rossum as the sole developer. Van Rossum has helped create the ABC language earlier in his career, which had notably more readable syntax than many other languages but also had a number of issues about it. Van Rossum took ABC as an inspiration, using a similar syntax and including the good features, but fixing the issues to make an even better language. The Python Software Foundation was established in 2001, to foster development in the Python community, including organising developer conferences and managing intellectual rights (Wikipedia contributors, Retrieved 2023). Named the “benevolent dictator for life” by the Python community, he retired from his position as chief decision-maker in 2019 (Wikipedia contributors, Retrieved 2023). A five-member Steering Council was elected to lead the project, with a new election occurring with each feature release (The Python core team and community, 2018).

Python is a multi-paradigm programming language supporting both object-orientated (using objects to contain data and code), and structured (control flow and subroutines for program clarity and quality) programming. Python also partially supports functional and aspect-orientated programming.

Python’s philosophy is based around the idea that simplicity is key and is summarised in The Zen of Python. Clarity and readability should be the foremost concern when writing code, with optimisation being performed when necessary and without impacting the original clarity. This emphasise of readability and minimalism is often referred to as “pythonic” (Wikipedia contributors, Retrieved 2023). “Pythonista” can be used to refer to a user of Python.

# Development Platforms

Python is compatible with a number of different development platforms, called integrated development environments (IDEs), as well as being supported across all main operating systems.

Popular IDEs include (Programiz, Retrieved 2023):

* IDLE - Python’s built-in default lightweight IDE
* PyCharm - a dedicated professional IDE specifically for Python
* Visual Studio Code – a free and open-source professional IDE for many languages

Instead of building all functionality into the core, Python was designed to by highly extendable via modules and libraries. This solved one of Van Rossum’s frustrations with ABC (Wikipedia contributors, Retrieved 2023).

The Python Standard Library contains a huge range of built-in libraries that can be easily imported into any Python project. This allows developers to easily access additional functionality outside of Python’s core, without the overload from hundreds of modules. These libraries contain functionality from additional datatypes and file formats, to networking communication and graphical user interfaces (Python, Retrieved 2023).

The Python Package Index (PyPI) is a third-party software repository run by the Python Software Foundation. It contains over 450 thousand packages developed and shared by the Python community (Wikipedia contributors, Retrieved 2023). Between the standard library and PyPI, there are a huge variety of extensions available to Python developers to have access to whatever functionality they may need.

# Characteristics

Python’s variables are strong and dynamically typed. Strong type variables can’t operate on items of different types, so concatenating a string and an integer will result in an error. Dynamic means that variables are assigned a type as runtime by the compiler or interpreter, based on the assigned value. This is in contrast to weak or statically typed variables (Wikipedia contributors, Retrieved 2023). Python also uses mutability, which means that most variable’s values can be altered after creation, however some data types are immutable (Great Learning Team, 2023). For instance, lists and dictionaries are mutable, however strings and tuples are not.

Python is a high-level language, which means that it has a strong abstraction from the details of the computer. This means that it has automatic memory management though garbage collection and uses a more natural language than lower level languages (Wikipedia contributors, Retrieved 2023).

# Conclusion

Python is an excellent language for those who are new or less capable with programming. It has beginner friendly syntax and a large community of support to help newcomers delve into programming. Python’s ease of writing, combined with its wide range of libraries and functionality means it can be a great choice for more professional and business applications too, such as data analytics.

It can however be slower than other languages, and less efficient as there is less control given to the developer on how memory is managed and other lower-level features. For small-scale programming this doesn’t matter too much, but with large-scale, intense programs, high memory usage is not ideal and will perform much slower. While there are extensions for web and game-based uses, it is also not designed specifically for these applications, so it is generally better to use other languages instead.

# References

Great Learning Team. (2023, August 14). *Understanding Mutable and Immutable in Python*. Retrieved from Great Learning: https://www.mygreatlearning.com/blog/understanding-mutable-and-immutable-in-python/#:~:text=An%20object%20is%20considered%20mutable,lists%2C%20dictionaries%2C%20and%20sets.

*History of Python*. (2023, November 1). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/history-of-python/

Programiz. (Retrieved 2023). *Python IDEs and Code Editors*. Retrieved from Programiz: https://www.programiz.com/python-programming/ide

Python. (Retrieved 2023). *The Python Standard Library*. Retrieved from Python.org: https://docs.python.org/3/library/index.html

The Python core team and community. (2018, December 16). *PEP 13 – Python Language Governance*. Retrieved from Python Enhancement Proposal: https://peps.python.org/pep-0013/

Wikipedia contributors. (Retrieved 2023). *Python (programming language)*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/w/index.php?title=Python\_(programming\_language)&oldid=1185632218

Wikipedia contributors. (Retrieved 2023). *Python Package Index*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/w/index.php?title=Python\_Package\_Index&oldid=1175539303

Wikipedia contributors. (Retrieved 2023). *Python Software Foundation*. Retrieved from Wikipedia, The Free Encyclopedia: https://en.wikipedia.org/wiki/Python\_Software\_Foundation