

|  |
| --- |
| SDV602 Assessment 2 |
| Final Reflection |
| Name: Rhylei Tremlett  ID: 13513865  Date: 19/11/2023 |

Table of Contents

[Introduction 1](#_Toc151318664)

[Heritage and Philosophy 2](#_Toc151318665)

[Platforms for Development 3](#_Toc151318666)

[Characteristics, Strengths and Weaknesses 4](#_Toc151318667)

[Conclusion 5](#_Toc151318668)

[References 6](#_Toc151318669)

# Introduction

Python is a popular and widely used programming language praised for its versatility, low learning curve, readibility, and supportive community.

Python was concieved in 1989 mostly on a whim by its prognietor Guido van Rossum. Python eventually was born and was built around the “Zen of Python” principles which dictate the values and direction of the language.

Python continues to gain more and more traction as artificial intelligence (AI) and machine learning (ML) become more popular and widely used, as Python is one of the dominant languages used in these fields. Accompanying AI and ML are big data and business analytics, both of which Python is also exceptional at, further increasing Python’s popularity and usage.

# Heritage and Philosophy

Python was originally developed in 1989 by Guido van Rossum during his two-week Christmas break. The development of Python started as merely a hobby for Guido. He spent much of his available time developing the language. Guido built every component of Python from the ground up. During his interview with Oxford Union, Guido says the following: “I built everything from scratch: the parser, the bytecode compiler, the object implementation, the long integers, and tuples, and strings, and everything.” (*Creator of Python Programming Language, Guido van Rossum | Oxford Union - YouTube*, 2019)

The first release of Python (0.9.0) was in 1991 and published as an open-source. The first major release (1.0.0) was in 1994. Python has had steady releases and improvements ever since.

Python is designed to be easily read and understood by everyone. Tim Peters created the ‘PEP 20 – The Zen of Python’, which contains 19 lines, with the final 20th line reserved for Guido to write. The Zen of Python embodies what the development team of Python strive to build and how developers should use Python (*PEP 20 – The Zen of Python | Peps.Python.Org*, 2004).

# Platforms for Development

Python is an interpreted language mean it doesn’t require compilation. Python is also hardware and operating system independent, Python creates bytecode which is interpreted by any python interpreter.

Python also has hundreds of development enviroments and editors. Some of the most popular include (in no particular order):

* Visual Studio Code
* PyCharm
* Jupyter Notebook
* PyDev
* IDLE

Python is very extensible due to its exensive package and module network ‘The Python Package Index’ (PYPI) The front page of PYPI describes itself as: “A repository of software for the Python programming language.” (*PyPI · The Python Package Index*, n.d.)

There are as of writing nearly 500,000 projects on PYPI, some of the most popular and frequently used are:

* Pandas
* NumPy
* TensorFlow
* Matplotlib

# Characteristics, Strengths and Weaknesses

Python is beloved by many developers for its readability and simplicity. Python has a straightforward and comprehenssible structure.

Python is also an incredibly versatile and flexible language, being a jack-of-all-trades language, Python has tools to create any kind of tool for any purpose.

Python is a dynamically typed language, meaning that variables can be declared as one type, then redefined later as another. However, Python also has strict typing, meaning that it allows for type hints and annotations to specify variable types and enforce type safety. Type hints facilitate code readability and inline documentation, aiding developers in catching potential errors before runtime. Python's strict typing also prevents implicit type conversion promoting clearer code and reducing unexpected behavior and errors caused by unintended type mixing.

Python is a high-level general-purpose multi-paradigm language, meaning it can do most things developers need it to do, however that doesn't mean it’s the most optimal choice for certain tasks. One example of this is mobile and desktop applications, there is very little usage and traction there, and because developers don’t want to use python there its not getting any better.

There is no perfect language, and Python is no exception. Python is not fast or efficient, due to it being dynamic and interpreted, placing more stress on the runtime compiler.

Python is memory intensive and a resource hog. Small python programs can use exponetially more resources than the same program written in a different language more optimized for the task. Python also has no multi-threading, instead imitating the effect with multiprocessing which relies on the operating system.

# Conclusion

Python stands as a testament to the power of community-driven open-source development and adaptability. Over 32 years, Python has evolved into a language revered for its simplicity, versatility, and extensive community support.

Python's evolution has experienced exponential growth, especially with the growth in artificial intelligence, machine learning and big data analytics, and business applications. The PYPI and Python’s platform independence have been critical in its widespread adoption across various domains.

Python remains the language of choice for many because of its ease of use, robustness, and the support of its community. Python’s journey shows the impact of a programming language built on shared principles, collaboration, and commitment to accessibility in the world of coding.

# References

*Creator of Python Programming Language, Guido van Rossum | Oxford Union—YouTube*. (2019, December 31). https://www.youtube.com/watch?v=7kn7NtlV6g0

*PEP 20 – The Zen of Python | peps.python.org*. (2004, August 22). https://peps.python.org/pep-0020/

*PyPI · The Python Package Index*. (n.d.). PyPI. Retrieved 19 November 2023, from https://pypi.org/