Software Development 2

*Milestone 3 – Python reflection*

Jackson Hayes

NMIT ID - 13521772

SDV602

S2 2023

Table of Contents

[Heritage & Philosophy 3](#_Toc150108629)

[Heritage: 3](#_Toc150108630)

[Philosophy: 3](#_Toc150108631)

[Environment 4](#_Toc150108632)

[Platforms: 4](#_Toc150108633)

[Code libraries: 4](#_Toc150108634)

[Characteristics 5](#_Toc150108635)

[JavaScript Comparison: 5](#_Toc150108636)

[Type system: 5](#_Toc150108637)

[Strengths & Weaknesses: 6](#_Toc150108638)

[Strengths: 6](#_Toc150108639)

[Weaknesses: 6](#_Toc150108640)

[References 6](#_Toc150108641)

# Heritage & Philosophy

## Heritage:

Python was created in the late 1980s by Guido van Rossum in the Netherlands as a successor to the ABC programming language aiming to fix handling and interfacing issues within the Amoeba operating system (Pramanick, 2023). The implementation of Python started in 1989 with a publicly available version on February 20, 1991, as Python 0.9.0. Python would then receive frequent updates throughout the years introducing new features like list comprehensions for example, in the Python 2.0 release during the year 2000.

The name “Python” was inspired by the British comedic group Monty Python with the desire for the language to be fun and accessible for users. Python was developed and initially designed as being a general-purpose language with its readability, simplicity and easy to learn style. This has made the Python language widely adopted by many I.T career paths including software development, engineering, machine learning, data analysis, web development, testing and more.

Key contributors for Python include:

1. **Guido van Rossum:**
   * The main author and contributor of Python until he stepped down from his role in 2018.
2. **Python Software Foundation (PSF):**
   * Founded in 2001 to support and protect python’s development. The PSF is mostly responsible for copyright, trademarks, and fundraising.

## Philosophy:

Python’s philosophy is often described in the “The Zen of Python”, a python enhancement protocol (PEP) that lists rules for writing and creating computer programs with Python. It enforces the core characteristics of the programming language previously mentioned which is simplicity and readability.

Python supports a wide range of programming paradigms including object-oriented (OOP), functional and procedural programming, giving a broader scope for problem-solving.

The maintenance and development of Python are handled by the core development team, volunteers and the Python community which contribute with PEPs. PEPS are used to propose changes to the programming language, submitted by any member of the community and then later reviewed by the developers before being implemented.

# Environment

## Platforms:

Python is supported on a range of platforms and operating systems including Windows, Linux, UNIX, macOS and others. Typically runs on server-side or any machine with the Python interpreter. The interpreter can be used on any operating system with the same codebase being usable across all platforms with little to no modifications.

## Code libraries:

The Python programming language has an extensive list of libraries available, some of the more popular ones include:

* **Matplotlib:** A plotting library used for the visualization and analysis of data that allows for creation of static, animated, and even interactive visuals.
* **NumPy:** Used for its mathematical functions, random number generators and algebraic routines, etc.
* **Pandas:** Used for its data analysis and manipulation of it. Provides ways to modify data sets like CSV files using Python code.
* **Django & Flask:** Frameworks incorporated for their tools related to web development.
* **Tkinter & PySimpleGUI:** Standard Python libraries providing an easy way to create simple GUI applications.
* **Requests:** A Python library for HTTP requests.

# Characteristics

## JavaScript Comparison:

JavaScript was the first programming language I learned with Python being the second language to properly use. Here is a brief comparison between Python and an existing programming language I have used.

Python’s philosophy is intended to be simple and readable, while JavaScript’s heritage is aimed at web development as a language for interactive websites. JavaScript is mostly used in web browser environments but can also be seen in server-side programming with the use of Node.js. Python on the other hand, performs well in a wide range of environments from server-side applications to desktop/educational.

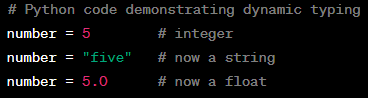
Other comparisons between Python and JavaScript include:

* **Type system:** Both Python and JavaScript are dynamically typed but additions to JavaScript like TypeScript offer static typing for JavaScript.
* **Syntax:** Python relies on indentation to define code blocks, while JavaScript uses curly brackets.
* **Concurrency:** JavaScript handles concurrency through callbacks and promise, with async/await methods. Python uses threads and code libraries.

## Type system:

Python uses a dynamically typed system, where the type of a variable is determined during runtime and doesn’t require specific declaration of the variable type. This means that the developer can redefine a variable later in the file without causing an error, which allows for faster software development but can lead to runtime issues that can be hard to identify.

An example of Pythons dynamically typed system:



## Strengths & Weaknesses:

### Strengths:

* **Versatility:** Wide range of libraries makes Python a multipurpose language that can be used for a variety of tasks from web development, server-side code and data analysis.
* **Syntax:** I find Pythons clear syntax allows for faster and easier development of applications.
* **Readability:** The clear syntax makes the code easier to understand and is quite intuitive
* **Community Support:** Python has a large community due to being one of the most used languages. This means that there is extensive support through documentation and forums.
* **Programming paradigms:** Supports a range of programming paradigms including object-oriented, functional, procedural, and more.

### Weaknesses:

* **Runtime errors:** Due to being dynamically typed, Python can through runtime errors that would usually be caught during compiling in statically type languages.
* **Mobile development:** A relatively small weakness but Python is not commonly used for mobile application development.
* **Memory:** Python consumes more memory in a trade off to being more flexible and easier to use.

# References

Pramanick, S. (2023, November 1). *History of python*. GeeksforGeeks. [https://www.geeksforgeeks.org/history-of-python/](https://www.geeksforgeeks.org/history-of-python/%20)

*Python® – the language of today and Tomorrow*. About Python. (n.d.). [https://pythoninstitute.org/about-python#:~:text=Python%20was%20created%20by%20Guido,called%20Monty%20Python’s%20Flying%20Circus.](https://pythoninstitute.org/about-python%23:~:text=Python%20was%20created%20by%20Guido,called%20Monty%20Python’s%20Flying%20Circus.%20)