SDV602 Assessment essay

Hamish Drogemuller 13515109

Contents

[Introduction: 2](#_Toc119692960)

[Heritage & Philosophy: 3](#_Toc119692961)

[Python Heritage: 4](#_Toc119692962)

[Javascript Heritage: 5](#_Toc119692963)

[Platforms for Development: 6](#_Toc119692964)

[IDEs: 6](#_Toc119692965)

[Code Libraries: 6](#_Toc119692966)

[Package Management: 6](#_Toc119692967)

[Python: 7](#_Toc119692968)

[Javascript: 8](#_Toc119692969)

[Code Comparison: 9](#_Toc119692970)

# Introduction:

This document aims to investigate and explain some of the differences between a language I am familiar with (Javascript) and a language that I have just learnt (Python).

# Heritage & Philosophy:

The history, ethos and role of a programming language greatly influence how a programming language behaves and reacts. To this end it is important to investigate and extrapolate information around this subject.

## Python Heritage:

### Heritage:

Python development began in the 1980s, developed by a man called Guido van Rossum, a dutch programmer. The first public release of the Python programming language was in 1991 versioned Python 0.9.0, Python 2 was released in 2000 and then an 8 year gap until we received Python 3 in 2008. These two versions were incompatible, Python 2 was maintained until 2020 when it was discontinued.

Python is considered a multi-paradigm programming language, including object-oriented, structured, functional and aspect-orientated programming. Other paradigms are available and supported through the use of extensions available to Python.

### Philosophy:

The core philosophies of Python are described within a document titled “The Zen of Python”, these philosophies being:

* Simple is better than complex
* Complex is better than complicated
* Explicit is better than implicit
* Beautiful is better than ugly
* Readability counts

A major factor of the popularity of Python is the fact that it is so expandable, this is achieved with the use of modules that allow the functionality of python to be expanded. The was the creator Guido van Rossum’s vision from the birth of the language, a small core language with a vast extensive library.

Pythons development is largely influenced through the Python Enhancement Proposal (PEP) documentation. This documentation is the main forum for the latest and major features that will be added. This documentation also provides stylised guides to ensure all developers (that wish to follow the PEP standards) have an easy to follow guide that keeps everybody on the same page.

## Javascript Heritage:

### Heritage:

Javascript was released in December 1995, it is a programming language that meets the ECMAScript standard. This standard is meant to ensure the synergy and compatibility of web pages across many different web browsers.

Similar to Python, Javascript is usable with multiple paradigms. These paradigms are event-driven, functional and imperative programming. Application programming interfaces (APIs) are usable within Javascript when using text, dates, expressions, data structures and Document Object Model (DOM). Currently, alongside HTML and CSS, Javascript is one of the major components of the web as we know it today. As of 2019 97% of websites utilised Javascript.

As of today, the current Javascript environment contains many libraries and frameworks that can be utilized by developers worldwide.

### Philosophy:

Javascripts ethos consists of a combination of lightweight self-implementation which is a lightweight version of an older object-oriented programming language known as Smalltalk. Meaning that the ethos was similar to its influenced language; Simplicity, Composability and Uniformity. Notably Javascript has evolved greatly since it was introduced to the world bringing it to the forefront of web technology today.

# Platforms for Development:

## IDEs:

IDEs or Integrated Development Environments are utilised to perform functions such as running and executing code, debugging, interpreting, compiling and acting as a terminal. It is known as a development environment that programmers utilise write, compile and debug code all in one place.

## Code Libraries:

Code libraries are collections of prewritten code that can be utilised by developers to optomise tasks. Usually these libraries provide specialised code that allows tasks to be completed easier. By utilising libraries, developers are able to spend a greater amount of time implementing their own visions of their application instead of having to build everything from scratch.

## Package Management:

Package management is conducted through the use of Package Managers, collections of software tools that automate the process of installing, updating, configuring and removing programmes from a computer system. The package manager is activated through a request from the user which prompts the manager to find and download the requested package. Once this is done it will then install and provide tips for any extra steps.

## Python:

Python is quite often used for the development of many applications such as server-side web development, software development, math, scripting, machine learning, data science and artificial intelligence. Python is popular for the fact that it can be used across a multitude of popular operating systems.

To make the most of Python there are a variety of IDEs that can be used such as:

PyCharm – This was the first IDE I was recommended when learning Python. It provides a specialised editor that assists python programmers with auto code correction, error detection, quick fixes and more.

Spyder – Open Source IDE that provides a powerful debugger greatly utilised in the scientific fields that require python.

Code Libraries can be used to extend the functionality of Python, some of these libraries include:

Tensorflow – Specially used in Machine Learning Algorithms, a very flexible library that has modular parts allowing for standalone operations. A major benefit of tensorflow is that since it is so popular there is a massive quantity of developers that can be turned to for more information.

NumPy – NumPy is simple, interactive and easy to use. It is used for expressing images, sound waves and other raw data. The large community is also a benefit with this library.

Pandas – A powerful library utilised when working with data. Allowing for additional functionality when working with a manipulating data. With a wide range of tutorials this is a great library to learn.

Package Managers are the key to expanding the functionality of our programming languages:

Pip Installs Packages (PIP) – is the primary package manager for Python, allowing the installation of additional libraries that can be used to extend the functionality of python.

## Javascript:

Javascript is one of the primary technologies behind web and application development, this is a large focus of many of Javascripts IDEs and libraries.

IDEs:

Visual Studio Code – My preferred editor that contains a vast array of features to facilitate development of applications. VS Code can be utilised to work with many other languages through the use of its vast library of extensions.

Atom – Open source and a predecessor to VS code, Providing a simplified suite of functionality.

Webstrom – Developed by JetBrains, they created an IDE with built in support for technologies and languages. Usable across many operating systems.

Code Libraries:

jQuery – A fast, lightweight and rich library developed in 2006 which is used to simplify HTML document manipulation and event handling.

React.JS – One of the predominant technologies used for front end development in our current age, Utilised to create UI elements.

Lodash – A Javascript library utilised for math operations and working with arrays.

Package Managers:

Node Package Manager (NPM) - Is the Javascript equivalent of PIP. It contains access to a staggering amount of libraries that can be used to enhance Javascript.

# Code Comparison:

|  |  |  |
| --- | --- | --- |
|  | Python | Javascript |
| Advantages | * Easy to learn – developed with an English-esque syntax making it easier to read and understand. * Free and Open Source * Large library support – Pythons extensive amount of libraries means any functionality can be implemented in an application. | * Speed – Javascript runs upon the client side, reducing back end load. * Popularity – Due to the popularity of JAvascript there is a wide range of information available if you become stuck. * Cooperative – Cooperates very well with other languages. |
| Disadvantages | * Slow – Execution conducted line by line, leading to slower execution. * Inefficient Memory usage * Runtime errors derived from the dynamically typed workings of the language. | * Security – Can be exploited for malicious intent due to it being a client side language * Browser Support – different browsers interpret Javascript differently, making it difficult to code an application that works everywhere. |

|  |  |  |
| --- | --- | --- |
| Code | Python | Javascript |
| Define | def stuff(x):  print(x)  Python uses indentation to define code blocks. This means that when a series of continuous lines of code are indented at the same level then they are part of the dame code block | function stuff(x) { console.log(x); }  JavaScript uses curly braces {} to group statements to the same code block to define their scope and use. |
| Variables | <variableName> = Value  x = 5  Python’s simplistic approach to define and declare a variable cuts down on code and is easy to read. | // var variablename = value ; var x = 5; let x = 5; The syntax just like Python is similar with JavaScript. The only difference is a key word is used before the name of the variable and a semicolon at the end of the value. |
| Naming Conventions | #first\_name  Python uses snake\_case naming style. Variable names follow the same conventions as function names | //firstName  JavaScript’s naming style is lowerCamelCase. This means that the name starts with a lower case letter then every new word starts with an uppercase letter. |
| Comments | #This is a comment  Python uses a # for comments. All characters in a single line after this are considered part of the comment. | //This is a comment  JavaScript uses two forward slashes // to start a single line comment |
| List vs Array | numbers = [1, 2, 3]  In Python, lists store a sequence of values in same data structure. They can be modified. | let numbers = [1, 2, 3] JavaScript, has an equivalent version called an array. |
| Logical Operators | # and or not  Python logic operators are written and easily to understand. | //&& || !  JavaScript uses symbolism to cut down on code. |
| Conditional Statements | # if condition: # code # else: # code  Like the define code block above Python has a simplistic approach that’s clean and simple to read. | // if (condition) { //code //} else { // code //}  JavaScript’s If, else statement is also simplistic in its design and shows the structure or outline of the code with {} curly braces. |

# Conclusion:

After investigating and comparing these languages it really showed me why I enjoyed python so much, the simplistic minimal mature of the language appeals to me personally. I can definitely see myself moving forward with Python more in the future.