Python as compared to JavaScript

Emily Friis

13505646

18/11/2021

# Heritage and Philosophies

### Python

Python as a programming language was first conceived in 1989 by Guido van Rossum as a successor to the programming language titled ABC. The first iteration of Python was released in 1991, eventually followed by the first, second, and third versions – released in 1994, 2000, and 2008 respectively. The programming language ABC, regarded the predecessor to Python, was considered to be designed as a simple, general-purpose scripting language. Python sought to expand upon the principles of ABC, taking up the following concepts as its major selling points (Tulchak et al., n.d.):

* Clean Syntax
* Tolerance Programs
* Normal Distribution featuring many useful modules
* Simple yet Powerful IDE with Normal Distribution
* Suitability for solving complex mathematical problems.

A succinct encapsulation of Python’s design philosophy was created by Tim Peters, an experienced Python lecturer, in the form of 19 Axioms labelled the “Zen of Python” (Python.org, n.d.):

*Beautiful is better than ugly.*

*Explicit is better than implicit.*

*Simple is better than complex.*

*Complex is better than complicated.*

*Flat is better than nested.*

*Sparse is better than dense.*

*Readability counts.*

*Special cases aren't special enough to break the rules.*

*Although practicality beats purity.*

*Errors should never pass silently.*

*Unless explicitly silenced.*

*In the face of ambiguity, refuse the temptation to guess.*

*There should be one-- and preferably only one --obvious way to do it.*

*Although that way may not be obvious at first unless you're Dutch.*

*Now is better than never.*

*Although never is often better than \*right\* now.*

*If the implementation is hard to explain, it's a bad idea.*

*If the implementation is easy to explain, it may be a good idea.*

*Namespaces are one honking great idea -- let's do more of those!*

### JavaScript

JavaScript as a programming language was conceived by Brendan Eich in 1995, developed as a simpler alternative to the Java programming language for Netscape 2. Originally title Mocha (later LiveScript), JavaScript was initially intended to provide versatile scripting and web development opportunities for amateurs, scripters, and designers working on web applications for the Netscape Navigator web browser (LEARN academy, 2020). Its initial release featured several of the concepts that led to its continued popularity and expansion to the current date:

* First-class Function Objects
* Java-esque Syntax
* Prototype-based Object Model

The general philosophy of JavaScript has changed greatly since its first inception, as it was passed from Netscape on to ECMA early in its history, causing a dramatic shift from having a small functional scope to a much larger one – seeking to expand its use cases and become a general-purpose language that can even be used to produce stand-alone applications through Node.js.

# Development Platforms

Solid development tools are crucial for being able to get the most out of any given programming language, regardless of how well designed it is. Both Python and JavaScript have a wide array of options when it comes to options to enhance their usability.

IDEs are especially important for programmers, as they provide the graphical interface through which coding is done with any given language – providing essential services such as:

* Saving/Reloading Source Code
* Built-in Compiling and Execution
* Debugging Support
* Syntax Support
* Automatic Formatting

Among other useful features that are becoming more prominent such as smart code completion accomplished through AI.

Code Libraries and Modules are highly reusable chunks of code used to expand the use cases for programming languages beyond their base functionalities from animation to complex maths.

### Python

Some notable development tools for Python include:

* IDLE is Python’s built in IDE, but is surprisingly well-suited for beginners to coding with Python. It does require the configuration of pip to install additional modules.
* PyDev is another Python IDE, that sacrifices some simplicity for more advanced development tools. It is functionally a plugin that allows Python and other adjacent programming languages to be use the Eclipse IDE (originally designed for Java).
* Several modules such as Matplotlib, Pandas, NumPy, and Requests act to support Python’s widespread applicability in server-side development, data analysis and more.

### JavaScript

Some development tools for JavaScript include:

* Visual Studio Code is an IDE that can be used for JavaScript and many other programming languages. It has an enormous market share compared to many other IDEs, granting it a large assortment of features and extensions to optimize user experience.
* React is a massively popular open-source framework for JavaScript web development. It can be used to great effect to create impressive dynamic user interfaces with solid performance.
* jQuery is a well-known JavaScript library that’s been around since 2006, being used for a wide variety of functions including DOM manipulation, event handling, animations, and more.

# Programming Environments’ Strengths & Weaknesses

### Python

Python’s greatest strength is as a high-level and general-purpose programming language focused on simplicity – accomplishing what other languages can but in a way that is more readable and using fewer steps. These features of its design make Python applicable across many projects – improving productivity substantially.

#### Strengths

* Simplicity – Python is one of the easier programming languages to learn as a beginner, optimizing the readability of its code with English-esque syntax, and requiring less coding overall.
* Bug Fixing – As an interpreted language, Python executes code line-by-line, stopping on errors and report them, making identifying issues with code very simple.
* Portability – Compared to many other languages, Python rarely needs changes to code in order to operate on different platforms, provided it doesn’t have any system dependencies.

#### Weaknesses

* Slow Speed – Python, while being really easy to learn through its nature as a dynamic and interpreted programming language, sacrifices a great deal of execution speed as a result.
* Poor Memory Efficiency – Python uses a lot of memory, consequently making it unreliable for use in client-side or mobile applications.
* Database Access Issues – Python’s database access layer is comparatively underdeveloped, making its use cases in enterprise-level projects fairly slim.

### JavaScript

Python as a programming language is hugely popular, particularly in the space of development for lightweight to medium sized web application – both in client-side and server-side. While it does require more concerted effort and a philosophical shift from simpler-to-understand languages such as Python, it is fairly well suited for its position in the programming industry.

#### Strengths

* Fast Speed – JavaScript is very fast by design. When it is run immediately within the client's browser, provided it isn’t waiting for a server response, JavaScript outpaces most other programming languages on the web.
* Server Load – Another benefit of reduced reliance on servers is that there will be much less demand for server resources using JavaScript.
* Support/Updates – JavaScript is widely used across the web, and is constantly receiving browser support for features such as ES6.

#### Weaknesses

* Client-Side Security Faults – In web applications, JavaScript code is executed on the client-side, leading to it being prone to exploitation for malicious purposes. In some cases, JavaScript may be disabled entirely to prevent loopholes in web security.
* Browser Support – Although being client-side yields great performance, not all browsers will interpret JavaScript code the same way – leading to some potentially unexpected outputs.
* Large Applications – JavaScript, while excellent for web development, is highly suboptimal for use in large applications, being unnecessarily difficult to code and maintain.

## Tabular Comparison

|  |  |  |
| --- | --- | --- |
|  | **Python** | **JavaScript** |
| Typing | Python is strongly typed and dynamically typed – data types are determined during runtime and can only change with an explicit conversion. | JavaScript is weakly typed and dynamically typed – data types are determined during runtime and can change implicitly. |
| Numeric Types | Python has integers, floating-point variables and complex numbers for numeric types. | JavaScript has floating-point variables as its only numeric type. |
| Mutability | Everything in Python is an object – types of objects are either mutable or immutable. | JavaScript has primitive and non-primitive types – primitive types are immutable, atomic data, and non-primitive types are mutable objects with multiple properties. |
| Execution Speed | Slower | Faster |
| Ease of Learning | Easier | Harder |
| Error Handling/Debugging | Easier | Harder |
| Object/Function Definition | Objects/Functions in Python are defined by indentation. | Objects/Functions in Python are defined by curly brackets. |
| Inheritance | Python uses Class-based Inheritance – classes inherit properties and methods from their relationship to their parent class. | JavaScript uses Prototype-based Inheritance – objects inherit properties and methods directly from a prototype object. |
| Encoding | ASCII by default | UTF-16 |
| Standard Library/Modules | Python’s standard library features a wide range of modules covering a large number of use cases. | JavaScript’s standard library contains very few modules (Date, math, regexp, JSON), but has access to a huge number of libraries through NPM. |
| Scientific Computing/Machine Learning | Python has a wide range of libraries dedicating to processing complex data and machine learning. | JavaScript is generally unsuitable for scientific computing and machine learning. |
| Mobile Development | Python is generally unsuitable for developing mobile applications. | JavaScript is suitable for developing mobile applications. |
| Web Development | Python is suitable for developing the server-side for web applications. | JavaScript is suitable for developing both the client-side and server-side for web applications. |

# References

Tulchak, L. V., Маrchuk, А. О. (n.d.) History of Python. Repository of Vinnytsia National Technical University. <http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/10471/461.pdf?sequence=3&isAllowed=>y

PEP 20 -- The Zen of Python. (n.d.). Python.org. <https://www.python.org/dev/peps/pep-0020/>

JavaScript history. (n.d.). W3Schools Online Web Tutorials. <https://www.w3schools.com/js/js_history.asp>

Brief history of JavaScript | Ruby on Rails vs JavaScript. (2020, February 10). LEARN academy - San Diego's Original Coding Bootcamp. <https://www.learnacademy.org/blog/who-introduced-javascript-originally-called-why-created/>