Tyler Cullen

Assignment: **Programming Languages I Might Like To Learn**

CSC 344 | Programming Languages

Abstract

This assignment presents six different programming languages which I might like to learn and why. Throughout the assignment I will research the programming languages origin, who contributed toward its development, where it stands in the programming landscape, and how it tends to be used. This will allow for me to grasp a better understanding of the programming language and perhaps motivate my classmates to look more into them as well.

---

1. C - 1972

The programming language **C** was developed by Dennis Ritchie in 1972. C is the successor to the programming language B which was also developed in Bell Labs by Dennis Ritchie. It was originally designed as a system programming language to write an operating system. C started gaining popularity in the 1980s and has become one of the most used programming languages to this day. (Wikipedia)

Why To learn C:

C has low-level access to memory, a simple set of keywords and a clean style gh making it great for system programming like operating systems or compiler development.

C reduces the gap between the low level and high level languages and helps you to understand the fundamentals of computer theories. This is because in most modern high level languages the machine level details are hidden.

Programs that are written dnc compiled in C are executed very quickly compared to other programming languages because they do not have processes such as garbage collection or preventing memory leaks.

---

2. Structured Query Language (SQL) - 1978

SQL was initially developed at IBM by Donald D. Chamberlain and Raymond F. Boyce after learning about the relation model from Edgar F. Codd in 1974 it would be officially released. SQL is a domain-specific language which is used for stream processing and data held and in a relational database management system. It is really well at handling structured data and creating relations among different objects and variables. (Wikipedia)

Why SQL:
SQL is valuable and is in high demand in many industries. Positions such as developers, business analysts, data science and more use SQL on a daily basis to create reports and find trends in their industry.

SQL is helpful for handling very large amounts of structured data. SQL makes it efficient to perform various operations such as getting rows based on certain criteria from large databases.

SQL is frequently updated to be friendly with new technology. Most of the data oriented technologies use a SQL interface. This makes it highly portable and efficient when working on any data or data driven technology.

---

3. Python - 1990

Python is a high-level general purpose programming language. It was developed in the late 1980s by Guido Van Rossum at Centrum Wiskunde & Informatica in the Netherlands as a successor to the ABC programming language. Python's key points are its code readability and use of indentation. It is dynamically typed and garbage collected. (Wikipedia)

Why Python:
Python has hundreds of libraries and frameworks due to its corporate sponsorship and supportive community. This allows for easy support and lots of solving common and difficult issues through the help of the community and libraries.

Python is efficient, reliable, and much faster then most modern high level languages and can be used in almost any kind of environment. Python can also be used on many applications such as web development, desktop applications, mobile apes, and more.

Python is great in automation of tasks. There are many tools, libraries, and modules which make things increasingly efficient for automation. Many programmers can reach an advanced level of automation using Python while also using a lot less lines of codes making it very efficient.

4. R - 1993

R was created by professors Rosh Ihaka and Robert Gentlemen as a programming language for students at the University of Auckland. R is an inspiration from the programming language S. R is used for statistical computing, data mining, bioinformatics and data analysis.

Why R:

R is an open source language making it maintained by the community and its active users. This is great because it allows for the programmer to edit, create their own packages and modify functions within R since it is registered under a public license there is no restriction of its usage.

R provides many ways to visualize data which is great for data analysis and data science which is a growing field currently. The fields of data scientists who are proficient in R make more than $117,000 on average per year making R one the musts in the higher payer job fields of data science.

R is the most comprehensive statistical analysis package since new technologies and ideas often appear in R first. R is also cross-platform which runs on many platforms and operating symptoms such as linux, windows, and mac.

5. RUBY - 1995

Creator Matsumoto claims he made Ruby in 1993 in a 1999 post to the “ruby talk” mailing list. Ruby is a high level general purpose programming language. Ruby was influenced by Perl, Ada, Java, and Lisp.Matsumopto describes the design of Ruby as being like a simple Lisp language at its core. (Wikipedia.)

Why Ruby:
Ruby is a great scripting language that developers use to automate manual processes. Because of this simple scripts can be written quickly and make things work efficient.

Ruby is often used in web scraping. Ruby has many libraries that are used to download and scrape web pages frequently. This allows for the analyzing use of libraries to further inspect the elements of a website/HTML.

Many programmers say that Ruby syntax allows for them to mold and shape their code whichever way they feel. Allowing them to think more about what they want to do and much less about how to construct it.

6. PHP

PHP is a general purpose scripting language geared toward web development. It was conceived by Danish-Canadian Rasmus Lerdorf in 19933 and released in

1996. PHP code is processed on a web server by a PHP interpreter and is often used for various web template systems and content management.

Why PHP:
PHP can do almost anything related to server-side scripting or the backend of a website. Such as receiving data from forms, generating dynamic page content, working with databases, and sending and receiving cookies and emails.

Applications can easily be loaded and connected to a database. This is due to its faster loading over slow internet speed unlike other programming languages. PHP applications can also run on any OS such as Windows, Linux, and UNIX.

It is stable with years of assistance providing continuous support to various versions. Helps powerful libraries to support the use of many different modules for data representation. PHP can be combined with many other programming languages in order to make the most effective use of the latest programming technology.