Could Python be replaced in the future?

Nicholas Robertson

Old Dominion University

CYSE 250

Professor Shobha Vatsa

December 5, 2022

Abstract

Seeing how everything changes over time it would be reasonable to say that the most popular form of computer language that everyone is using today, will be outdated in the coming years and that's what this paper is about. This paper is meant to compare and contrast Python, the most popular coding language on the planet, with other languages, most notably Java and Java Script, to see what the benefits are and draw backs between them. It's supposed to ultimately answer the question, is there a possibility that Python will be replaced in the future by another coding language? Though my findings I've come across compelling arguments on both sides. Some praise Python for its widespread usages, while others praise Java Script for it's already established framework and familiarity with people in the AI development fields. But ultimately the findings came down to the simple fact that Pythons versatility and accessibility was what kept people so satisfied with the language. It was deemed for how easy it was to use and for how many jobs/careers favored Python, that it would most likely never be replaced even decades after the writing of this paper. Currently there is nothing that comes close to competing with it.

Key Words: 1- IoT 2- import socket 3- soc=Socket.Socket() 4- print
5- soc.bind 6- soc.listen 7- while True 8- conn.addr=soc.accept()
9- conn.send and .encode 10- conn.close 11-soc.recv and .decode

Introduction

Python is the most universal and easy to learn languages for computer learning out there. It is responsible for building “websites and software, automate task, and conduct data analysis.” (Coursera, 2022). But while Python is very useful there is a lingering fact that prosiest in every form of technology, whether it’s existed in the past, present, or future. The fact that eventually everything has the possibility to be replaced in the distant future. This fact doesn’t just exist with physical hardware rotting and deteriorating from the passage of time, but it also extends to digital media and software as well. As something new and interesting comes along and replaces the old forms of entertainment, a similar argument can be made about programing languages, with the factors being easier and more versatile to use in the ever-evolving world we live in. The question then becomes, what can possibly replace Python and what would the new language need to have to be able to replace a widely used and popular codding language.

Possible candidates

For possible replacement options you must consider a few key factors that make Python a favorable option. Factors like versatility and having it be easy to use play a big role in the popularity of a langue and how well it will translate into the working field. So, with that in mind if you take a language like Java, which is the third most popular programing language according to the website Truelist. It’s praised for its versatility, and it is seen as the best for [1] internet of things (IoT), which is a term used to explain how certain software can be used with any device that has the capability to connect to the internet. For instance, Java can be run on any computer out on the market, which is one of the main selling points of Java and its why people tend to gravitate towards it. It is also said that “Java is the official programming language for Android development, with Java accounting for 46.2 percent of all Android applications.” (Fran, 2022). So, the advantage of accessibility is there, but the main draw back with Java, that Python has a leg up on is the way it can be used. For instance, Python is good for coding artificial intelligence and even website development, but Java is

much more focused on enterprise grade applications, which is mainly used in the development of apps. So, seeing this we can discount Java as a successor to Python as it's firmly planted in the application side of coding. There is also JavaScript, which in terms of website development, JavaScript is better as "Java Script runs in the browser while Python is backend ser-side language." (Anna Johansson). Which means, in basic terms, that Java Script can be used to develop websites stand alone without the need for backend server-side like Python does. A backend server-side "consists of the server which provides data on request" (Concepta Technologies, 2022). Java Script can also be used for AI development just like Python, but the interesting thing about this though is that "Java Script is more commonly used due to the fact that most developers already know it to some extent." (Xccelerate, 2019). Essentially what this means is that the people going into the field of AI development should in a sense already have some sort of background in using some form of Java over Python, but this also sheds light on Python's accessibility though, as it can still be used as an easier to learn alternative in AI development that Java Script. Pythons simplicity is it's saving grace for machine learning, as "its clear code, machine learning-specific libraries and possibility to shift focus from the language towards algorithms." (Nazar Kvartalnyi, 2022). We can look at newer emerging languages that might have a shot at taking pythons place, but I just wanted to set the tone and give some examples of the most likely already established languages that might have had a chance in taking its place. For now, let's turn our attention to the types of people that use Python to conduct their work.

Public Appeal

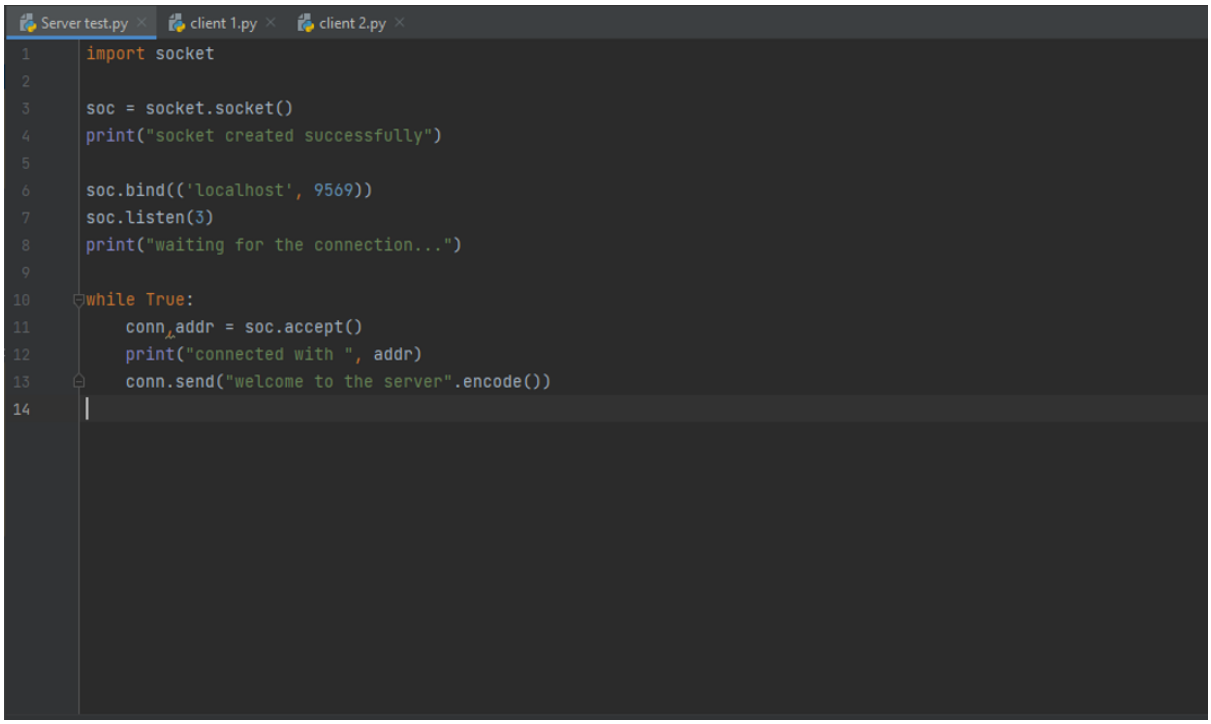
When looking at reasons on why Python is more popular with people learning computer language, we must think about the types of people that tend to use these coding tools to begin with. Python's easy to learn language system isn't just convenient, it's quintessential to the overall functionality of the language to begin with. It's what grabs people's attention because, it doesn't require you to have a deep background in mathematics or to already be familiar with HTML and CSS. This means more people, whether they're at a college, high school or thinking about learning at home on their down time as a hobby, it's possible for them to pick it up and make a full fledge career out of it. This also brings me to another point about how many applications and jobs in cybersecurity and coding relies on Python, making the already easy to learn coding language that much more enticing to pick

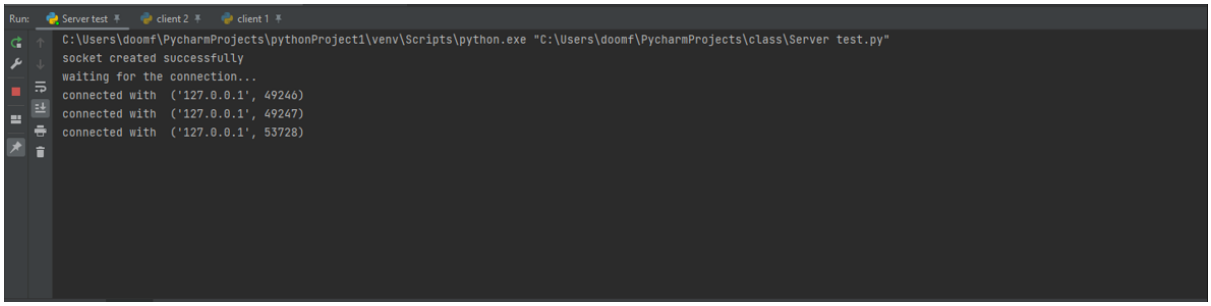
up. It should be stated that the other languages mentioned before are not necessarily bad, in fact there are some skills set out there like web creation and application creation, in which languages like Java are far more appealing thanks to some quality-of-life improvements that they have over Python. The simple truth is that, if the language is easier to learn, people are going to look over some of the short comings that might come along with it. Taking this into account, let's see how easy it is to create a server in python.

Creating a Server in Python

To create a server and client within python, you first want to start with the server side. Doing so will guarantee a stable connection within, so that the client can connect. You want to start with typing in the first line [2] import socket within the first line. What this command does is that it will send a message across a network. Afterwards, you want to type [3] "soc = Socket.Socket() in the second line, this function creates the socket. Then in the third line you want to type, [4] "print("socket has been created successfully"). The print function prints out the message put in parentheses to signify the successful creation of the socket. In the fourth line type [5] "soc.bind(('localhost', 8979)). This command gives the socket a name, while the string of numbers at the end gives the client a connection code for them to connect to. So if the server has the numbers 8979 then the client must also have 8979 in order for them find the server and connect. In the fifth line you want to type [6] in soc.listen(3). This function asks as a listening device to see who's going to connect to the that specific server and the number at the end indicates how many clients are allowed to connect to it. Then in the sixth line you want to type in print("waiting for the connection…"), which will indicate a connection is trying to be made to the server from the clients. In the eight line you want to type in [7] while True:, this will initiate a loop forever, which will allow the statements made with in the while True statement to be repeated until the socket is closed. In the ninth line you want to type in [8] conn,addr = soc.accept(). Which will accept sockets from clients that are attempting to connect to the server. In the tenth line type in print ("connected with", addr) to signify that the clients are connected to the servers address and then in the eleventh line type in [9] conn.send("welcome to the server".encode()). This command will send out the message written in parentheses to the clients connected to the server. Then finally type in [10]

conn.close(), to close the socket. Below will be a example image taken from my computer to give visual reference to the instructions.





This second screen shot is a example image of the clients connecting on the server side.

Client side

For the client side, you have the ability to connect with the server and depending on how many clients the server allows, depends on the soc.listen command that was in the servers code. This connection not only allows you to receive messages from the server, but it also allows you to send and receive messages from other clients. To start off, it is very similar to the server side, by using import Socket in the first line and soc=Socket.Socket() in the second line. In the third line your going to want to type “soc.connect((‘localhost’,8979)). Then in the fourth line type, [11] “print(soc.recv(1024).decode()). This function is just stating that the message from the server side was received on the client’s end. Doing this will allow you to connect with the server host and be able to receive messages from them.

Findings

I found that in most if not all cases, Python was found to be the most versatile, easy to learn and most widely used coding language out there. While others could have been the in line to becoming the ones to replace Python, their limitations and complications set them back from becoming something more. When researching on this topic, I started to realize that no matter what the other languages might have had inters of advantages against Python, there was always one thing that Python was better at then them. For instance, as explained in the Java Script section of the paper, with AI coding, Java Script had a clear advantage over Python and considering everything else it seemed to be the better option. But the advantage that Python always seem to have, is its accessibility. That freedom to learn this language, thanks to the vast library of information available to the pubic is the main driving force that makes this language much more appealing.

Conclusion

In conclusion I believe that Python as of now, or even in the distant future, will not be replaced by anything. It has such a wide range of uses and is so easy to learn, that I am convinced that this language that will not even fade away for decades to come. It will go down in history as one of the best programing languages ever created by man and will help code the future of AI and ML (machine learning). I might not be able to predict the future, but I can almost certainly predict that this coding language will remain the most used and most accessible language even fifty years into the future.

References

1. Coursera (2022) “What is python used for? A Beginner’s Guide,” – *coursera.org* https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

2. Fran (2022). “What is Java used for?,” -*Futurelearn.com* https://www.futurelearn.com/info/blog/what-is-java-used-

for#:~:text=Java%20is%20the%20official%20programming,IDE%20for%20developing%20Android%20apps.

3. J. Anna . "5 Reasons JavaScript is still better than python," – *IEEE Computer Society*, *computer.org* https://www.computer.org/publications/tech-news/build-your-career/5-reasons-javascript-is-still-better-than-python

4. What is the difference between front-end and back-end development – *Conceptatech.com* https://www.conceptatech.com/blog/difference-front-end-back-end-development#:~:text=The%20back%2Dend%2C%20also%20called,interacting%20with%20the%20front%20end.

5. Xccelerate (2019). "Python vs JavaScript: Which Language you should learn and why," – *xccelerate.medium.com* https://xccelerate.medium.com/python-vs-javascript-which-language-you-should-learn-and-why-cd29c64935c

6. K. Nazar (2022). "Why use python for machine learning?" -*inoxoft.com* https://inoxoft.com/blog/why-use-python-for-machine-learning/