Adrian Rivera
Jorge Alonso
Víctor Navas

JAV Programming Language

February 26, 2021

Introduction

Python is a programming language that's simple to understand yet very powerful. It allows one to focus more on implementing code without the hassle of understanding the tool (Python) to a high level. Similarly we want to create a programming language similar to Python: functional and easy to use.

However, the user will still have to learn the syntax for the language and stick to it. We would like to allow more flexibility syntax wise in our language so a programmer can code with custom syntax rules that they think is optimal for the program. For example, if one wants to write a web application with our language they might follow ES6 standards while someone writing a kernel driver uses C/C++ standards and their respective keywords/operators.

Another advantage of this is that it can reduce length of code considerably. Take for example Java's keyword for inheritance **extends**, one may consider it not only just a lengthy word but also perhaps in the context of their code it doesn't make much sense. The programmer may choose to change this to **from** or a single character such as **~**.

It is up to the programmer to decide the style of coding for their project and this language aims to allow such flexibility. There will be a pre set of words that are used in most of the languages like **if** cases and **for** loops which will help in the simplicity of the programming language.

Language Features

One important aspect that we want our language to implement is the use of custom keywords and operators, where a programmer can specify what symbol/character to use. For example, let's say the default way to write comments in the language is using the `#` character the programmer can choose to override it to something else, such as `//`. This can be declared through either a config file or at the top level part of the main file.

A feature such as this allows for many different coding patterns to be viable. A C++ dev can write code in this language and share it with other C++ programmers, while a Python programmer can do the same. Since we're allowing flexibility in coding styles, an approach similar to Typescript could be convenient. Where a programmer can opt in for strict type checking or not. Just like the custom keywords/operators, a programmer can specify their choice. Then if type checking is on, things such as ` int string = "String" ` would cause a compile error or at the very least a warning.

Just like in Python, we would like to have a built-in list implementation so the programmer doesn't have to worry about simple data structures. Besides we will also include common mathematical operators within these lists. Like for example add all the numbers inside the list or with another list, create a set from the unique values, etc.

Example of a program

```
from pip._vendor.distlib.compat import raw_input
from statistics import mode
from statistics import median
import array as arr

overload comments as //

ans = True

// Example overloaded comment
while ans:
        print("""
        1.Add Data
        2.Calculate Statistics
        3.Log Out
        """)
        ans = raw_input("What would you like to do? ")

        if ans == "1":
        input_string = input("Enter data separated by space ")
        Numlist = input_string.split()

        print("\n Data Added")

        elif ans == "2":

        sum = 0.0
        subs = 0.0
        avg = 0.0

   for num in Numlist:
        sum += int(num)
        subs -= int(num)
        avg = sum / len(Numlist)
        model = mode(Numlist)
        midNum = median(Numlist)

   print("Sum = ", sum)
print("Substraction = ", subs)

print("The average number is ", round(avg))

print("The mode of the list: ", model)

print("The median of the list: ", midNum)

print("\n Calculation complete")


    elif ans == "3":
    print(Numlist)

    print("\n Goodbye")

    ans = None

    else:
    print("\n Not Valid Choice Try again")
```

Implementation requirements and tools

We will be using Python for this project since it allows us to use both functional and object oriented programming patterns. Moreover, Python allows us to focus more on designing and implementing the language without putting much effort into Python's syntax since it's simple, powerful and extensive. We will need an application like PyCharm to create this project since it's very extensive and it has a lot of features. For the group project files we will be using GitHub since it helps with the organization of all the files and versions of the project.

Project plan and timeline

Weekly stuff:

- Every wednesday: team meeting
    - Point to go over:
        - What have we implemented
        - Features to be added
        - Language features
        - New Ideas

2/14/2021 - Introduction
2/14/2021 - Language Features
2/20/2021 - Example of Program

2/20/2021 - Implementation requirements and tools
2/26/2021 - First Phase done
3/5/2021 - Lexical analyzer
3/12/2021 - Syntax analyzer
3/19/2021 - Intermediate Code
3/23/2021 - Second Phase done