**Language Overview and Setup**

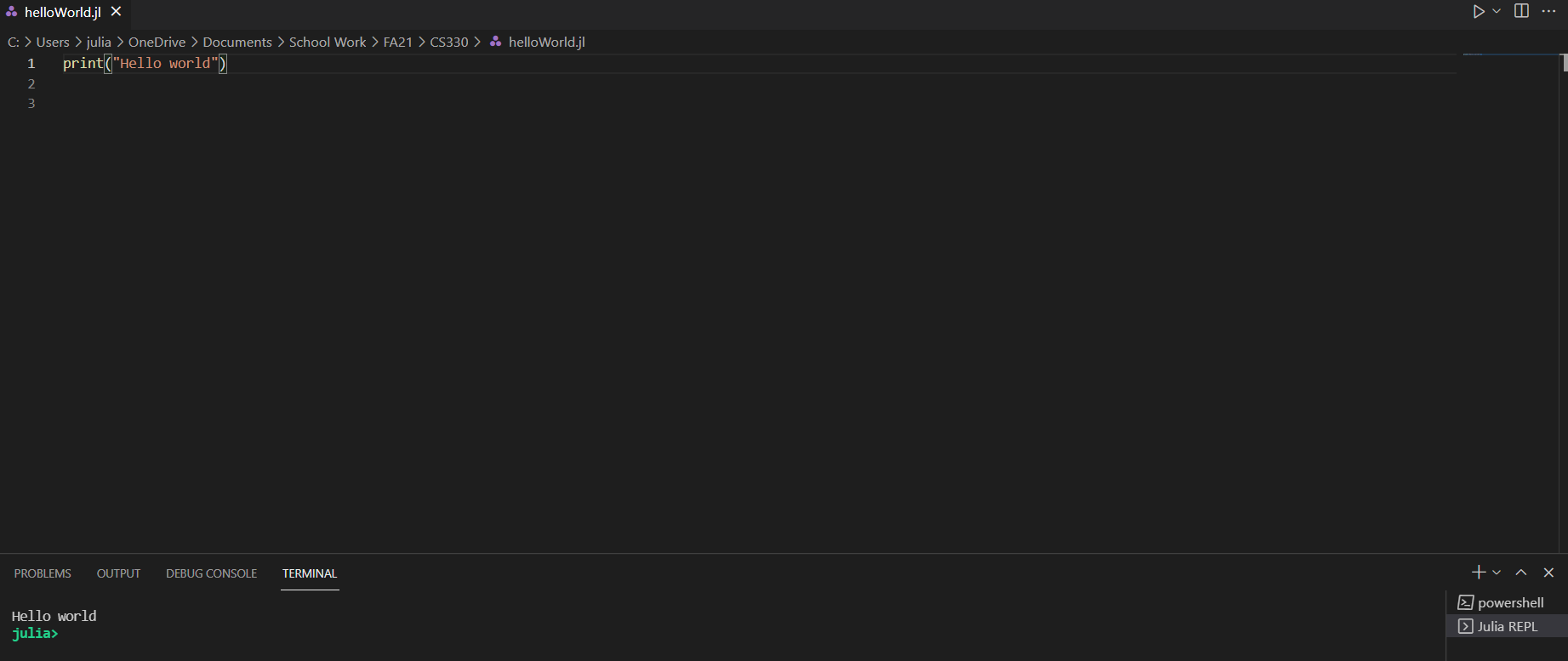
**History**

1. The name of my programming language is Julia.
2. Julia was created by Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman of MIT, and was initially released in 2012. In a 2012 blog post titled “Why We Created Julia”, the creators state that they created the language because they were “greedy.” They wanted a language that combined all of the best parts of other programming languages and sought out to make it. Some of the specifics they wanted to include are an open source license, a fast language like C, a dynamic language like Ruby, a language that could handle mathematical notation like Matlab, and many more useful features of many more programming languages. This makes Julia a mishmash of all different programming languages, and able to perform many different tasks (Bezanson et al., 2012).
3. Julia was developed to be fast and useful for statistics and data science, especially when dealing with big data. However, it’s similarities to Python allow it to be used as a general purpose language (*Julia Language Introduction).*
4. When it is time to begin programming in Julia, I will get information from the documentation page first. The documentation for Julia seems to be very complete. There also seem to be tutorials from geeksforgeeks.org, which I trust for syntax help. If I need more help beyond this, the website for Julia has pages dedicated to resources for learning Julia, including a list of books and YouTube videos.

**Getting Started**

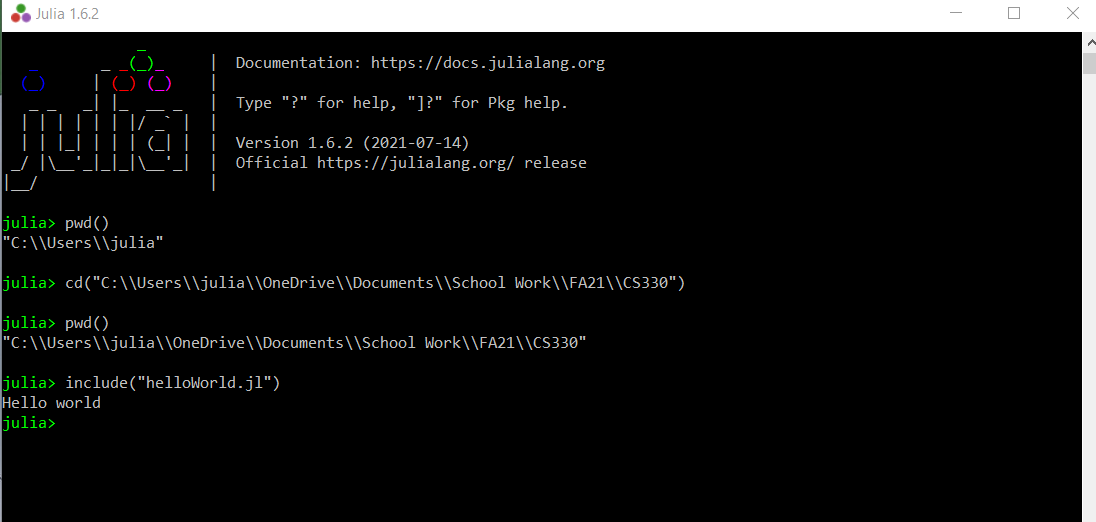
1. To install Julia, I downloaded the version for my OS from the Julia website (<https://julialang.org/downloads/>). I also had to add Julia to the path correctly to get programs to run from the Julia REPL (interactive command line) or my computer’s command line.
2. Julia can be run in a few different ways. I chose to add an extension to Visual Studio Code since that’s what we normally use in class, but a regular text editor can also be used as long as program files are saved with the correct extension, .jl. It seems that a lot of different IDEs support Julia extensions, according to a list *Top 8 IDEs and editors for julia*. I am not considering the rankings mentioned here, but this list shows that there are many options. Julia can also be used directly from the command line or it’s own REPL, but I prefer to code in an IDE.
3. To run programs using the extension in Visual Studio Code, you just have to hit the play button and the output will appear in the terminal. From the command line, change the directory to where your program is saved and type “julia programName” (not in quotes, just used for demonstration). From the Julia REPL, change the directory to where your program is saved and type “inclue(‘programName’)”, with the program name in quotes.

Running from VSC (*Julia Language Introduction)*:

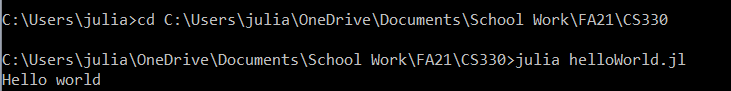


Running from the Julia REPL

(<https://stackoverflow.com/questions/22240581/running-julia-jl-files>):



Running from the command line (<https://stackoverflow.com/questions/42030150/how-to-exectute-julia-code-in-command-line>):



1. Single line comments in Julia are written just as they are in Python, with # followed by the comment. A multi-line comment in Julia is started with #= and ends with =# (*Comments in Julia*).

**Sources**

Bezanson, J., Karpinski, S., Shah, V. B., & Edelman, A. (2012, February 14). *Why we created julia*. The Julia Programming Language. https://julialang.org/blog/2012/02/why-we-created-julia/.

*Comments in Julia*. GeeksforGeeks. (2020, February 27). https://www.geeksforgeeks.org/comments-in-julia/.

*Julia language introduction*. GeeksforGeeks. (2020, April 16). https://www.geeksforgeeks.org/julia-language-introduction/.

*TOP 8 IDEs and editors for julia*. Dunebook. (2020, February 17). https://www.dunebook.com/top-8-ides-and-editors-for-julia/.

**Future Resources**

* <https://julialang.org/learning/books/>
  + List of books directly from the Julia page
* <https://www.youtube.com/user/JuliaLanguage>
  + Link to the Julia YouTube page