SDV602 Project

Documentation & Essay

Celeste Quinn
ID: 13508189

Table of Contents

Documentation

Login/Register

The program uses a login and registration service to maintain a user database. If for some reason, registration of a new user is not working, you can login by using this user:

u: celeste
p: no

Uploading a CSV & Displaying

You must do this in a strict order or it may crash or not run as intended. First click the browse button on the main window to open the file browser, than search for the CSV file “nz convictions (copy).csv” and upload it. Then click the display graph button and it will show. Said csv file is uploaded in the directory of the project. This program is built to run using this csv file only and will not work with others.

Chat

You must be on the active window to use the chat. It takes a very long time for messages to send for some reason. If you navigate to a different window, you cannot use the chat in another window until you press prev or next for the window you want to talk in.

Navigation

Prev and next buttons are what shifts control to each window. You have to remember yourself what button you clicked and what window is the main focus. Clicking the buttons again will not bring the window to the top focus if they are all open, but you MUST click the button before you can interact with. Interacting with a window that is open, but you havent clicked prev or next to shift control to, will not work and may cause unexpected errors in the program. For example, you have opened windows 1,2, and lastly 3. Now you want to get back to window one and chat. You must click prev on window 3, then prev on window 2, to be able to interact with window 1 again. In short, this is very annoying and touchy.

Essay

Python and C#

Python and C# are both object orientated languages however Python is dynamic and C# is static typed. Python was developed in the 1980s by a guy called Guido van Rossum, as his hobby project over winter. The same guy also created a language called ABC and based it loosely off that, fixing the problems that ABC had and borrowing the syntax and some features. It was built to be readable to to enhance developer productivity. It is now open source, so continues to be developed by programmers around the world, which is unique as far as languages go as most of them are developed by companies rather than individuals. Python took its name from Monty Python’s Flying Circus.

C# was developed by Microsoft in 2000 with the goal of being a general purpose object language, this means it can do more things than functional or procedural languages can do. C# has been upgraded along the years with different versions coming out that implemented more features into the language. It went from v1.0 all the way to v10. It was originally created to compete with Java, made by Sun, later Oracle.

Choosing an IDE and libraries for your language should be based upon what exactly you want to create, what features you will need, basically just what your plans are for it. An IDE is an Integrated Development Environment, more than just a basic code editor, the IDE comes with features such as debuggers or other things to help run programs, rather than just a simple place that you can write code. Libraries are like things you can import into your IDE that comes with functions to streamline a process, to be able to do more than just the built-in stuff that comes with the language. There are some IDEs that are specifically made for Python, such as Pycharm and Spyder. These two are written in Python and are both cross-platform. Pycharm is considered as the best IDE for using Python with, and Spyder can integrate easily with the main Python libraries used for data analysis. You can also use a generic IDE such as Visual Studio like we use at NMIT, that supports a whole range of different languages. As far as libraries go, well these are open source so can be developed by anyone. There are over a hundred thousand python libraries. The way you pick those would be again, depending on what you want out of your application, how you need it to work. There is probably a library that can help you with anything that you want. Matplotlib and Numpy are two such libraries that we have used, and have all kinds of features to be able to manipulate data, specifically graphs and charts, and working with numbers.

With C#, you can also use Visual Studio to write with as its Microsofts proprietary software. I'm not sure that there are any others that are made specifically for C# like there is with Python though. MonoDevelop is another good one. This comes with the game development program Unity, as an inbuilt IDE, so if you're going to use C# for coding games, then it would be handy to have that seamlessly integrated into Unity. C# requires .NET framework to run, while I'm not sure if this counts as a library for this particular point, well it's a whole framework that it uses and supports the language, and it's huge. WindowsForms is another big one that it uses, and this can be access visually within Visual Studio, to create fullout windows screens by drag and drop within the IDE. I'm sure there would be more libraries for C# than there are Python.

C#

characteristics

-simple
-modern
-component oriented
-fast

strengths

- garbage collector to stop memory leaks, which was a problem in C++
- its made so you can write minimal code to accomplish tasks
- its a legible and terse language, so it makes sense to someone else just by looking at it, making it easier for teams or more than one person to write

weaknesses

- garbage collector can use a lot of memory in large applications
- it can be complicated for beginners when faced with private, public, protected, etc. I found when I first used it, it was a nightmare to try to access certain functions from other functions or from other scripts and ended up being a big mess.

Python

characteristics

-dynamic
-easy
-open-source
-object oriented

strengths

-extensible, python can have new functions added onto it, and can interface with libraries written in other languages
-can program with a GUI using tkinter
-interpreted language. This is helpful for something like debugging because you know the exact line where you messed up is where it'll stop.

Weaknesses
-slower than other languages because it is interpreted
-higher memory consumption

Conclusion

What I liked a lot about python was something I noticed, I could name a variable 'global' within a definition and then easily access it outside of its definition, just like that, no hassle. The same with importing definitions from other files, you can do that so so easily with python, however C# it is not so easy as I went into above. You have to mess around making the entire class protected or something and then override it and then just go through so much effort trying to access things from other scripts, I actually think I burnt that out of my memory as a bad time.

I feel like after this class I am definitely no expert on python and have barely any experience actually using it, as my stuff was mostly with the pysimplegui library, it was difficult for me to differentiate those bits of code with python itself. I would love to be able to make a comparison here from personal experience, but the whole process of learning it was difficult for me and not as easy as all the articles will tell you about python being easy to pick up and learn.

There are many differences between the two anyway, C# is definitely more complex and has more layers and features and things you have to know how to do before you can just jump in and start it, whereas python apparently this is not so.

References

Pramanick, Sohom. 2019. *History of Python.* Accessed at: https://www.geeksforgeeks.org/history-of-python/

Advani, Vaishali. 2020. *Open source python libraries you should know about*. Accessed at: https://www.mygreatlearning.com/blog/open-source-python-libraries/