

## Learning Journal

### Exercise 1.1:

1) Frontend development is the design and implementation of the parts of an application, program etc. that an end user would interact with. Simply put its all of the things that a user will see and use, be it graphics, text, pictures etc., backend however is the unseen by an end user structure of any running program. Normally this will consist of things like server management endpoint usage and routing among other things. I like to think of backend as all of the code that runs when a user presses a button, and the frontend is the code for the button itself (although it can apply to much more than buttons).

2) For me the difference between python and javascript has always lied primarily in the usability / readability of everything. Python in a lot of ways has always felt like the way I would explain how a coding language works ie: c++ or javascript. I think this mainly comes from the simplicity of the syntax as it feels much more similar to an actual spoken language than most other programming languages ever will. If I were to convince a team to use python, my best argument would be based on that same feeling as I feel it allows you to make code much more easier to understand, and you might think as programmers they should understand regardless, but as someone who has seen blocks of code span multiple monitors it all starts to jumble very fast. Specifically the variable access and management allows you to quickly alter and reassign values of the same name in a way java simply does not.

3) Goal 1 for me would be to become MUCH more comfortable with CLI. I have a large amount of experience coding in a "standard" environment, managed by editors and other applications, but I have always felt CLI practices never came natural to me. Goal 2 would be to change my point of view on python as personally I have always felt python was a bit almost discombobulated for me. I never truly felt at home coding in it, and at multiple points throughout college I felt it was different for the sake of being different with no actual improvement in mind. Obviously I know there are benefits to python, I just haven't quite yet found what those are for me. Goal 3 would be to familiarize myself with python libraries. When I selected this class as a specialization I did some research into python and found that there are vast amount of libraries used that make python much more applicable to a career

### Exercise 1.2)

1). I think the main benefit for me of the lpython shell is just the ease of use. I have always disliked writing with CLI just solely based on the horrible readability of it. All outputs being essentially the same color and spacing as inputs has always bothered me, and iPython is much more similar to a normal editor

2) String : array of chars : non-scalar

    Int : integer : scalar

    Bool : true or false value: scalar

    Float : number decimal points : scalar

- 3) A list is a mutable tuple, meaning the elements can be altered
- 4) I would do the same thing I did in my cookbook, with the individual words being stored as dicts, since there are multiple values each one needs to hold, and then using a list as a catalog of all our dicts, allowing us to change the order and add or remove if necessary.

#### Exercise 1.3)

1)

```
Destination = str(input("Enter your desired Location"))
Place1 = "cabo"
Place2 = "prague"
Place3 = "geneva"
If destination == place1:
    print( "Enjoy your stay in Cabo")
Elif destination == place2:
    print( "Enjoy your stay in prague" )
Elif destination == place3:
    print( "Enjoy your stay in geneva")
Else:
    print("Oops, that destination is not currently available")
```

2) Logical operators give us the option to perform multiples of checks at the same time. Whether that is checking one variable for multiple things, or checking multiple variables for one, with them we have the option to combine what could be a multitude of lines all into one.

3) Functions allow us to create our own essentially baby script within a file. It gives us the opportunity to run the same processes based on our own parameters and gives us the ability to use them repeatedly without having to reuse the same code over and over again while keeping things very consistent.

4) I feel as if I have become a bit more familiar with CLI mainly just navigating, however with everything im doing being "new" in python it has still been a bit unwieldy. I still feel mostly as if python is more awkward, the learning we have done so far is very basic and I knew most of it previously so my view hasn't been shifted much. At this point we have not really used any libraries so no progress there.

1.4)

- 1) If you weren't able to store files, all of the work done would be essentially pointless as none of the data would remain once the terminal is closed

- 2) A tool that converts data types commonly used in python into readable binary data. You would use them at any point where you are migrating data out of one file and need to perform operations on that same data in a different file.
- 3) `os.getcwd()`   `os.chdir()`
- 4) Using error handling block like `try except`, then to prevent it from breaking you would designate a resolution for that error, for example in this task we assigned the data an empty list when an error occurred so that the program will not end when there is no data
- 5) I feel like I am in good spot with what we have covered so far. Most of it has been more of a refresher but some of it has been new.