**IT5016 Assessment 3 - READ ME**

Reflective Journal

Over the last few weeks learning python I think my coding ability has improved significantly – I still have a way to go before gaining confidence and a lot more practice especially with PEP8 standards and beginning to simplify.

Starting out my codes were not fool proof, if an invalid or even capitalized letter was inputted there was no back up plan, it would simply fail. A lot of testing helped me get past that phase; however, I think sometimes I over complicated things and went a little OTT (Over the top) in this stage.

First beginning a lot my codes adhered to the **KISS** ("keep it simple stupid") principle particularly the sleep calculator, it wasn’t necessarily pretty in the terminal but behind it was very simple, cohesive, and easy for someone to read.

In one instance in my practice python code, I wanted to practice a greeting and attempt my functions, using this function improved the codes readability and I think adhered to the **DRY** (don't repeat yourself) principle by simplifying what could’ve been long print (“”) text.

I also felt in this code that by including “if” and “else” statements it was early-stage testing ensuring any input added was logical and made sense, even though it was a feature added prior to my function I believe this was an example of the **YAGNI** (“you aren’t gonna need it”) principle. This is because while I coded for functionality that was certainly going to be needed, I knew that it was something I wanted to account for to make my code more practical.

In my assessment I remember having to change the way menus were displayed, this was as it was a repetitive feature I was using, by applying the **KISS** methodology I not only shortened the code, but also made for a more practical code with improved readability.