Dunder and Magic methods

We require the use of Dunder and MAgic methods to help us identify ,use and output parameters of our own.

Most of the time a class that has no Dunder would return the location of memory the values were store within

Of the Dunder and Magic methods many exist such as :

\_\_init\_\_:

This method helps us define the Obj that will be used within our class as constructors.

If the values we set in \_\_init\_\_ aren’t set when we call the class this would cause the class calling to fail and an error message to pop requiring we input correct number of parameters

\_\_str\_\_:

This method allows us to return string instead of memory location, we use this by using

Print(ClassName)

\_\_setitem\_\_ & \_\_getitem\_\_ & \_\_delitem\_\_:

These three methods directly help with dictionaries within classes as \_\_setitem\_\_ allows us to set key or value of a dictionary within Class.

\_\_getitem\_\_ similarly allows us to return values within a dictionary of they exist.

\_\_delitem\_\_allows us to remove a Key and value within a dictionary if they exist.

\_\_len\_\_:

This method helps with getting the length of a dictionary or List if used with our class

Also determine the number of content we each list or dictionary have

\_\_contains\_\_:

Method is used to check if our class contains a certain element, the trigger for this dunder method seems to be unique as Class needs to be called within a If -- in statement

Encapsulation and Abstraction

Encapsulation and Abstraction seem to be a way of operating and a practice while working on programs, the process includes separating the program to many parts depending on how the program function as whole and provide comments and giving variable/functions/classes names related to their usage so that other programmers can understand the general operation of each code.

It is seen when we create a program it is better if we have access amount of data to enable many functions each fro storing data but split so that the function that stores important data to the program and the system is working separately from function that also stores data but not of great importance, it is always good to abstract the required data before entering any unnecessary data.

Asterisk arguments and Double Asterisk key arguments

\*arg & \*\*kwargs

They are many used with functions when we have no idea how many parameters we may need our function to accept at a time

\*arg is usually used to indicate variable or data structure

\*\*kwargs represent dictionary and we can even call \*\*kwarg.items()

When \*arg is set we can expect any number of variable and they enter the function in the form of tuples.

When \*\*kwarg we can expect Key words to be used along side values, Key words are set as dictionary keys and values are set as dictionary