Python/Database API

class DB:

The DB class deals with connecting to the database. It represents the connection to the database and contains all relevant functions.

Functions:

__init__(self, hostName, userName, password, dbName)
This constructor takes the host name, username, password, and database name as arguments and connects to the database.

add(self)
This function adds an entry with the current time to the database.

Fields:

self.db
This variable deals with the actual connection to the database. It requires the host name, username, password, and database name. These fields need to be manually inputted into the file. If any of these files are inputted incorrectly, when the connection tries to proceed, it will throw the exception error.

self.cur
This object creates the cursor that can be used to navigate through the database.

self.cur
These objects represent a database cursor, which is used to manage the context of a fetch operation. Cursors created from the same connection are not isolated, i.e., any changes done to the database by a cursor are immediately visible by the other cursors. Cursors created from different connections can or can not be isolated, depending on how the transaction support is implemented (see also the connection's .rollback() and .commit() methods).

add_part
This variable is used for inserting into our database. We use a similar style as MySQL when adding into a database with python. We make sure that the fields are lined up in the proper order. The %s is used as a placeholder for any type of value.

currentDT
This variable is used to grab the current time from the computer to be used in adding it into the database when the motion sensor detects motion. It grabs the date, day of the week, and time from the computer. This soon gets filtered for our needs for the database.

data_time

We use this variable to then take the currentDT variable and filter it to only display the clock time. We get the hours, minutes, and seconds from it.

self.cur.execute(add_part, data_time)
With this object, we then tell the cursor to our database to execute the command of inserting into the database. It knows to do this since the variable add_part has the string MySQL code to insert into a database. The cursor will then take our info and put it into the database.

self.db.commit()
This object method just ensures that all changes to the database are saved.