JSON to SQL Approach

|  |
| --- |
| 1. **Install sqlite in virtual environment**    1. conda install sqlite 2. Download static file into repository with parse\_Json.py file 3. **Open file:**   with open(json\_info) as jsonfile:  data = json.load(jsonfile)  This creates a list with a dictionary inside. See below. To get into the list, you must first index the list. E.g. print(data[0]) will produce the first dictionary inside. In the example below, this would be only Smithfield North. print(data[1]) would produce the entire dictionary then for Parnell Square North. This includes all fields between the {} brackets.     1. **Convert data to sql database**    1. Need to convert the key values into variables: i.e. address, latitude, etc. So first create empty list variables with the appropriate names.    2. Iterate through to the end of the json\_file and append the empty lists with the indexed values.   i = 0  while i < len(data):  number.append(data[i][*"number"*])  name.append(data[i][*"name"*])  address.append(data[i][*"address"*])  latitude.append(data[i][*"latitude"*])  longitude.append(data[i][*"longitude"*])  i += 1   * 1. **Create a new database.**   # create database to parse data into.  con = lt.connect(*'db\_Dataset.db'*)     * 1. **Create table for database.** (REAL is float in sqlite)   with con:  cur = con.cursor()  cur.execute(*"CREATE TABLE Static(num INT, Sname TEXT, addr TEXT, lat REAL, long REAL)"*)   * 1. **Loop through list variables and populate the new database**   for i in range(0, len(data)):  cur.execute(*"INSERT INTO Static (num, Sname, addr, lat, long) VALUES(?,?,?,?,?)"*,  (number[i], name[i], address[i], latitude[i], longitude[i]))  con.commit()   * 1. **Close database connection and cursor**   cur.close()  con.close()   * 1. **Print out dataset.**       + Connect to database, load all the data.      + Fetch all the data loaded      + Print the data      + Close the connections.   con = lt.connect(*'db\_Dataset.db'*)  cur = con.cursor()  cur.execute(*'SELECT \* FROM Static'*)  data2 = cur.fetchall()  pprint(data2)  cur.close()  con.close() |

Other useful information

|  |
| --- |
| **Download SQLITE Browser**  <http://sqlitebrowser.org/>  DB Browser for SQLite is a high quality, visual, open source tool to create, design, and edit database files compatible with SQLite.    **Tutorial for SQLITE**   * Inserting variables to database table - SQLite3 with Python 3 part   <https://www.youtube.com/watch?v=qfGu0fBfNBs>   * Read from (SELECT) Database table - SQLite3 with Python 3 part 3 <https://www.youtube.com/watch?v=NCc5r7Wr7gg> |