1. Set the variable test1 to the string 'This is a test of the emergency text system,' and save test1 to a file named test.txt.

In [1]:
```
test1 = 'This is a test of the emergency text system,'
print(test1)
with open('test.txt','w') as file:
    file.write(test1)
    file.close()
```

```
This is a test of the emergency text system,
```

In [2]:
```
! type test.txt
```

```
This is a test of the emergency text system,
```

2. Read the contents of the file test.txt into the variable test2. Is there a difference between test 1 and test 2?

In [3]:
```
with open('test.txt','r') as file:
    test2 = file.read()

print(test2)
print(test1 == test2)
```

```
This is a test of the emergency text system,
True
```

3. Create a CSV file called books.csv by using these lines:

title,author,year
The Weirdstone of Brisingamen,Alan Garner,1960
Perdido Street Station,China Miéville,2000

Thud!,Terry Pratchett,2005
The Spellman Files,Lisa Lutz,2007
Small Gods,Terry Pratchett,1992

In [4]:
```
data = '''title,author,year
The Weirdstone of Brisingamen,Alan Garner,1960
Perdido Street Station,China Miéville,2000
Thud!,Terry Pratchett,2005
The Spellman Files,Lisa Lutz,2007
Small Gods,Terry Pratchett,1992'''

with open('books.csv','w') as file:
    file.write(data)
```

In [5]:
```
!type books.csv
```

```
title,author,year
The Weirdstone of Brisingamen,Alan Garner,1960
Perdido Street Station,China Miéville,2000
Thud!,Terry Pratchett,2005
The Spellman Files,Lisa Lutz,2007
Small Gods,Terry Pratchett,1992
```

4. Use the sqlite3 module to create a SQLite database called books.db, and a table called books with these fields: title (text), author (text), and year (integer).

In [6]:
```
import sqlite3
db = sqlite3.connect('books.db')
cursor = db.cursor()
cursor.execute("CREATE TABLE books (title text, author text, year int)")
db.commit()
db.close()
```

5. Read books.csv and insert its data into the books table.

In [7]:
```
import sqlite3
import csv
conn = sqlite3.connect("books.db")
cursor = conn.cursor()
with open("books.csv","r") as file:
    books = csv.DictReader(file)
    for book in books:
        cursor.execute("INSERT INTO books VALUES (?,?,?)",(book['title'],book['author'],book['year']))
conn.commit()
conn.close()
```

6. Select and print the title column from the books table in alphabetical order.

In [8]:
```
import sqlite3
conn = sqlite3.connect('books.db')
cursor = conn.cursor()
output = cursor.execute("SELECT title FROM books ORDER BY title ASC")
for ele in output:
    print(ele[0])
conn.commit()
conn.close()
```

```
Perdido Street Station
Small Gods
The Spellman Files
The Weirdstone of Brisingamen
Thud!
```

7. From the books table, select and print all columns in the order of publication.

In [9]:
```
import sqlite3
conn = sqlite3.connect('books.db')
cursor = conn.cursor()
ouput = cursor.execute("SELECT * FROM books ORDER BY year")
for record in ouput:
    print(record)
```

```
('The Weirdstone of Brisingamen', 'Alan Garner', 1960)
('Small Gods', 'Terry Pratchett', 1992)
('Perdido Street Station', 'China Miéville', 2000)
('Thud!', 'Terry Pratchett', 2005)
('The Spellman Files', 'Lisa Lutz', 2007)
```

8. Use the sqlalchemy module to connect to the sqlite3 database books.db that you just made in exercise 6.

In [10]:
```
import sqlalchemy
conn = sqlalchemy.create_engine('sqlite:///books.db')
conn
```

Out[10]:
```
Engine(sqlite:///books.db)
```

9. Install the Redis server and the Python redis library (pip install redis) on your computer. Create a Redis hash called test with the fields count (1) and name ('Fester Bestertester'). Print all the fields for test.

In [11]:
```
! python -m pip install redis
```

```
Collecting redis


[notice] A new release of pip available: 22.2.2 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip


  Downloading redis-4.3.4-py3-none-any.whl (246 kB)
     ------------------------------------ 246.2/246.2 kB 3.7 MB/s eta 0:00:00
Requirement already satisfied: packaging>=20.4 in c:\users\dell\appdata\local\programs\python\python38\lib\site-package
s (from redis) (21.3)
Collecting async-timeout>=4.0.2
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Collecting deprecated>=1.2.3
  Downloading Deprecated-1.2.13-py2.py3-none-any.whl (9.6 kB)
Collecting wrapt<2,>=1.10
  Downloading wrapt-1.14.1-cp38-cp38-win_amd64.whl (35 kB)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\dell\appdata\local\programs\python\python38\lib\sit
e-packages (from packaging>=20.4->redis) (2.4.7)
Installing collected packages: wrapt, async-timeout, deprecated, redis
Successfully installed async-timeout-4.0.2 deprecated-1.2.13 redis-4.3.4 wrapt-1.14.1
```

In [18]:
```
import redis
conn = redis.Redis(host='127.0.0.1', port=6379)
conn.hset('test',{
    'count':1,
    'name':'Fester Bestertester'
})
conn.hgetall('test')
```

...

10. Increment the count field of test and print it.

In [19]:
```
conn.hincrby('test', 'count', 1)
conn.hget('test', 'count')
```

...

In [ ]:
```

```