**ADITYA AMIN ASSIGN : 20**

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.

test1 = 'This is a test of the emergency text system,'

# Save test1 to test.txt

with open('test.txt', 'w') as file:

file.write(test1)

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

Yes, there is a difference between test1 and test2. test1 is a Python string variable that contains the value 'This is a test of the emergency text system,', whereas test2 will be a Python string variable that contains the contents of the file test.txt after reading it.

# Read the contents of test.txt into test2

with open('test.txt', 'r') as file:

test2 = file.read()

# Compare test1 and test2

if test1 == test2:

print("test1 and test2 are the same.")

else:

print("test1 and test2 are different.")

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

import csv

# Data to be written to the CSV file

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']

]

# Write data to books.csv

with open('books.csv', 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(data)

print("books.csv has been created.")

1. 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).

import sqlite3

# Connect to SQLite database (or create if not exists)

conn = sqlite3.connect('books.db')

c = conn.cursor()

# Create books table

c.execute('''CREATE TABLE IF NOT EXISTS books

(title TEXT, author TEXT, year INTEGER)''')

# Commit the transaction and close the connection

conn.commit()

conn.close()

print("books.db database and books table have been created.")

1. Read books.csv and insert its data into the book table.

import sqlite3

import csv

# Connect to SQLite database

conn = sqlite3.connect('books.db')

c = conn.cursor()

# Read data from books.csv

with open('books.csv', 'r') as file:

reader = csv.reader(file)

next(reader) # Skip header row

for row in reader:

title, author, year = row

c.execute("INSERT INTO books (title, author, year) VALUES (?, ?, ?)", (title, author, year))

# Commit the transaction and close the connection

conn.commit()

conn.close()

print("Data from books.csv has been inserted into the books table in books.db.")

1. Select and print the title column from the book table in alphabetical order.

import sqlite3

# Connect to SQLite database

conn = sqlite3.connect('books.db')

c = conn.cursor()

# Execute SQL query to select title column in alphabetical order

c.execute("SELECT title FROM books ORDER BY title ASC")

rows = c.fetchall()

# Print the title column

print("Titles in alphabetical order:")

for row in rows:

print(row[0])

# Close the connection

conn.close()

1. From the book table, select and print all columns in the order of publication.

import sqlite3

# Connect to SQLite database

conn = sqlite3.connect('books.db')

c = conn.cursor()

# Execute SQL query to select all columns from books table ordered by year

c.execute("SELECT \* FROM books ORDER BY year ASC")

rows = c.fetchall()

# Print the selected columns

print("Books ordered by publication year:")

for row in rows:

print("Title:", row[0])

print("Author:", row[1])

print("Year:", row[2])

print()

# Close the connection

conn.close()

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

from sqlalchemy import create\_engine

# Define the SQLite database URL

db\_url = "sqlite:///books.db"

# Create an SQLAlchemy engine to connect to the database

engine = create\_engine(db\_url)

# Test the connection by executing a simple query

result = engine.execute("SELECT 1")

if result.fetchone()[0] == 1:

print("Connection successful!")

else:

print("Connection failed.")

# Close the SQLAlchemy engine

engine.dispose()

1. 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.

import redis

# Connect to Redis server

r = redis.StrictRedis(host='localhost', port=6379, db=0)

# Create Redis hash 'test' with fields 'count' and 'name'

r.hset('test', 'count', 1)

r.hset('test', 'name', 'Fester Bestertester')

# Get all fields from 'test' hash

fields = r.hgetall('test')

# Print all fields and their values

print("Fields and values for 'test' hash:")

for field, value in fields.items():

print(f"{field.decode('utf-8')}: {value.decode('utf-8')}")

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

import redis

# Connect to Redis server

r = redis.StrictRedis(host='localhost', port=6379, db=0)

# Increment 'count' field of 'test' hash by 1

r.hincrby('test', 'count', 1)

# Get updated value of 'count' field from 'test' hash

count = r.hget('test', 'count')

# Print updated value of 'count' field

print(f"Updated value of 'count' field: {count.decode('utf-8')}")