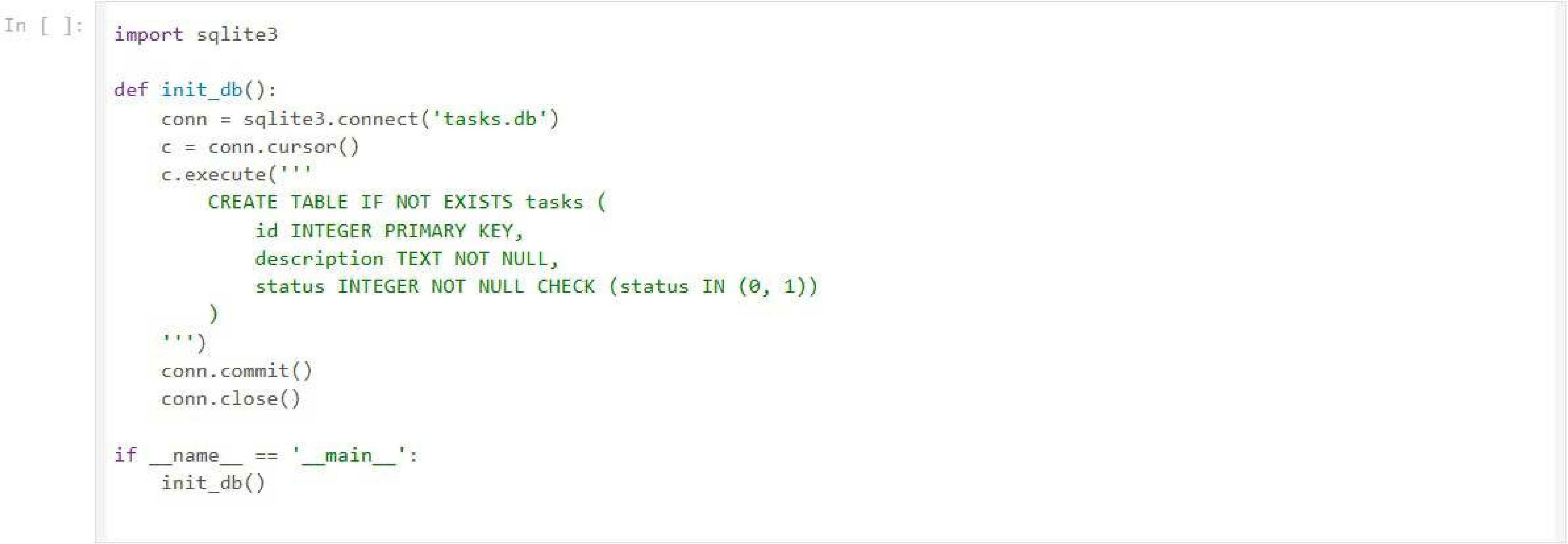


In [ ]:

```python
import sqlite3

def init_db():
    conn = sqlite3.connect('tasks.db')
    c = conn.cursor()
    c.execute('''
        CREATE TABLE IF NOT EXISTS tasks (
            id INTEGER PRIMARY KEY,
            description TEXT NOT NULL,
            status INTEGER NOT NULL CHECK (status IN (0, 1))
        )
    ''')
    conn.commit()
    conn.close()

if __name__ == '__main__':
    init_db()
```

In [ ]:

```
from flask import Flask, request, render_template, redirect, url_for
import sqlite3

app = Flask(__name__)

def get_db_connection():
    conn = sqlite3.connect('tasks.db')
    conn.row_factory = sqlite3.Row
    return conn
```

In [ ]:

```
@app.route('/')
def index():
    conn = get_db_connection()
    tasks = conn.execute('SELECT * FROM tasks').fetchall()
    conn.close()
    return render_template('home.html', tasks=tasks)
```

In [ ]:

```
@app.route('/create', methods=('GET', 'POST'))
def create():
    if request.method == 'POST':
        description = request.form['description']
        status = request.form['status']

        conn = get_db_connection()
        conn.execute('INSERT INTO tasks (description, status) VALUES (?, ?)',
                     (description, status))
        conn.commit()
        conn.close()
        return redirect(url_for('index'))

    return render_template('create.html')
```

In [ ]:

```
@app.route('/update/', methods=('GET', 'POST'))
def update(id):
    conn = get_db_connection()
    task = conn.execute('SELECT * FROM tasks WHERE id = ?', (id,)).fetchone()

    if request.method == 'POST':
        description = request.form['description']
        status = request.form['status']

        conn.execute('UPDATE tasks SET description = ?, status = ? WHERE id = ?',
                     (description, status, id))
        conn.commit()
        conn.close()
        return redirect(url_for('index'))

    return render_template('update.html', task=task)
```

In [6]:

```


def delete(id):
    conn = get_db_connection()
    conn.execute('DELETE FROM tasks WHERE id = ?', (id,))
    conn.commit()
    conn.close()
    return redirect(url_for('index'))
```

In [16]:

```
!pip install flask
```

In [16]:

```
!pip install flask
```

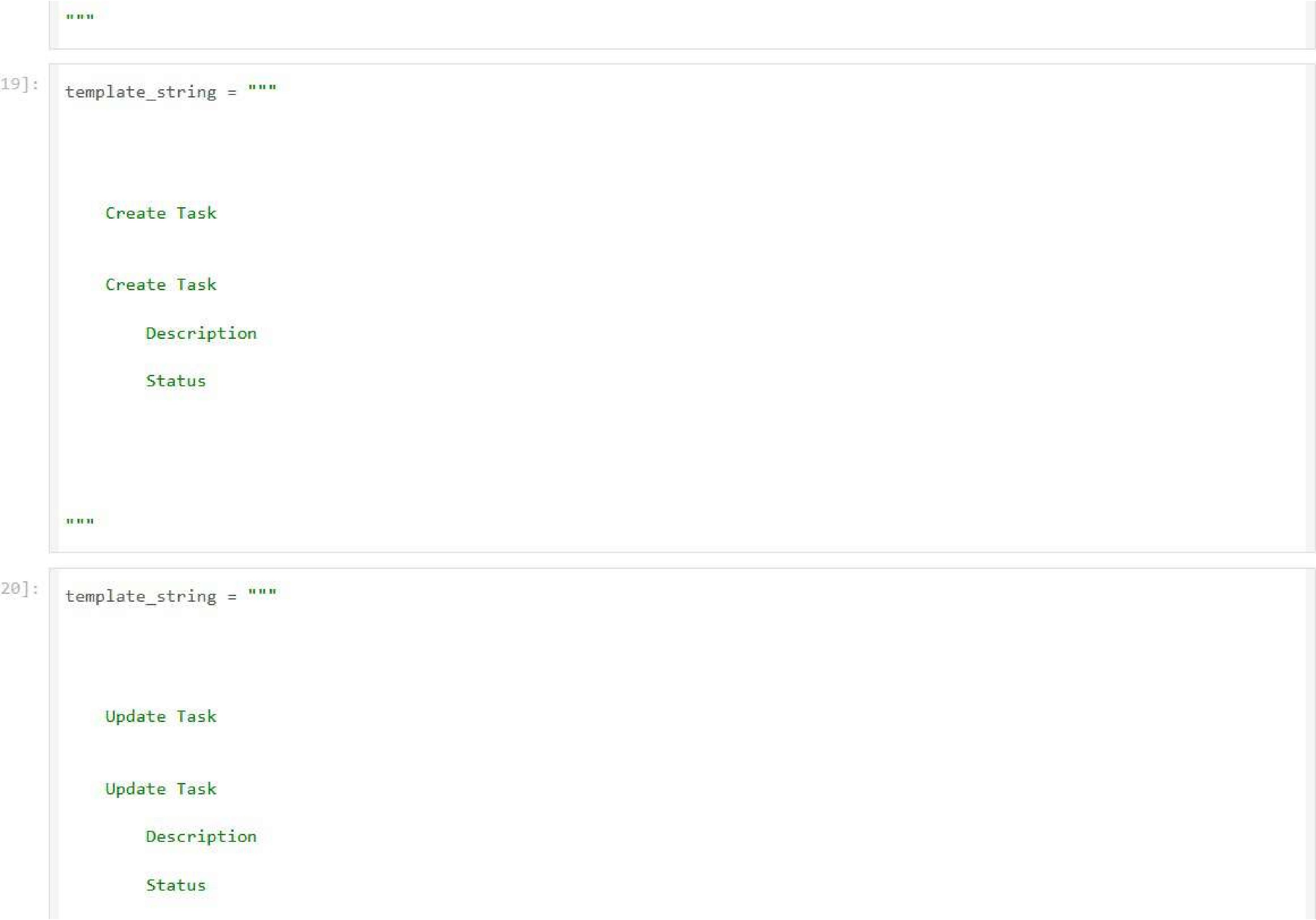
```
Requirement already satisfied: flask in /usr/local/lib/python3.10/dist-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from flask) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in /usr/local/lib/python3.10/dist-packages (from flask) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in /usr/local/lib/python3.10/dist-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in /usr/local/lib/python3.10/dist-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in /usr/local/lib/python3.10/dist-packages (from flask) (1.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=3.1.2->flask) (3.0.2)
```

In [27]:

```
from jinja2 import Template
```

In [18]:

```
template_string = """




    Task List


    Task List
    Create New Task

        {% for task in tasks %}

            {{ task.description }} - {% if task.status == 1 %}Complete{% else %}Incomplete{% endif %}
            Edit




        {% endfor %}
```



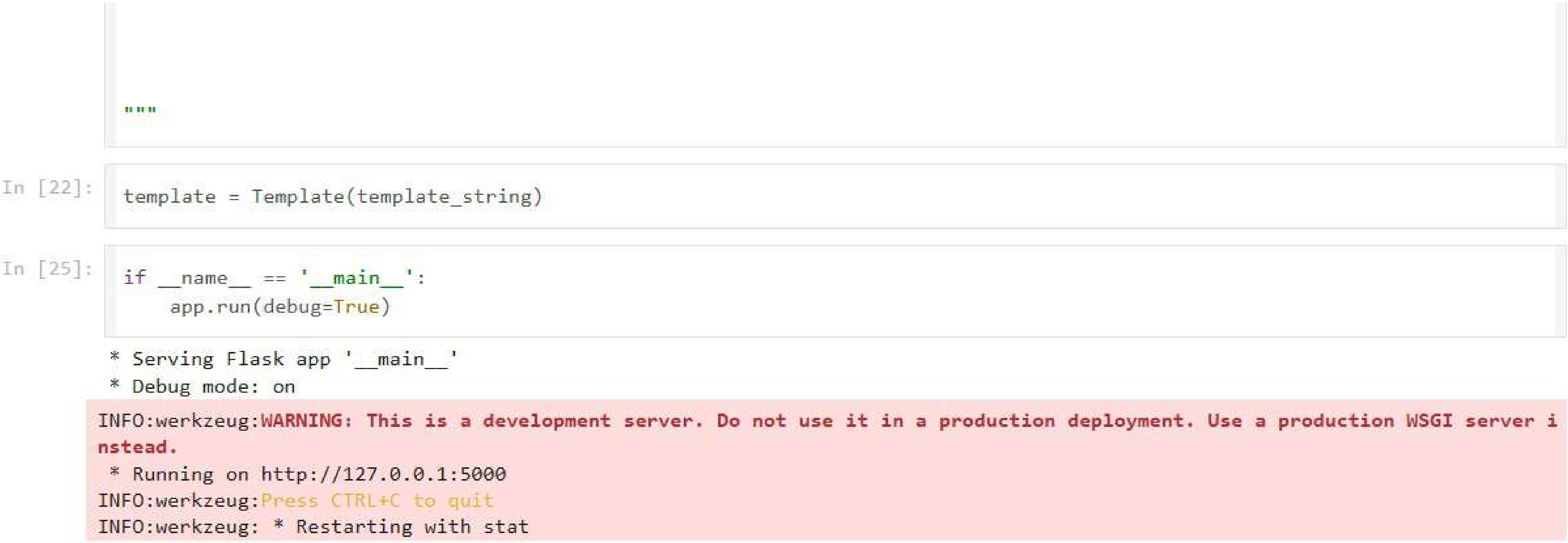
```
"""
```

19]:

```
template_string = """



    Create Task


    Create Task

        Description

        Status




"""
```

20]:

```
template_string = """



    Update Task


    Update Task

        Description

        Status
```



```
"""
```

```
In [22]: template = Template(template_string)
```

```
In [25]: if __name__ == '__main__':
             app.run(debug=True)
```

```
 * Serving Flask app '__main__'
 * Debug mode: on
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server i
nstead.
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug: * Restarting with stat
```