To-Do List Project Documentation

# Overview

This is a small project for managing a to-do list. The idea is simple: users can add tasks, view their tasks, mark them as completed, and remove them if needed. It’s a lightweight program, meant more for learning than for production use.

# Features

* Add new tasks with a short description.
* View all current tasks in a numbered list.
* Mark tasks as 'done' (but still keep them in the list with a status).
* Delete tasks completely if they’re no longer needed.
* Optionally, save tasks between sessions (depends on implementation).

# How It Works

1. When the program starts, it either loads an empty list or reads tasks from a file (if saving is enabled).
2. The user is presented with a menu (e.g., Add / View / Complete / Delete / Quit).
3. Based on input, the corresponding action is performed.
4. Changes are either kept in memory or written to a file before quitting.

# Example Flow

1. Add task  
2. View tasks  
3. Complete task  
4. Delete task  
5. Quit  
Choose an option: 1  
Enter task description: Buy groceries  
  
Task added!  
  
Tasks:  
1. [ ] Buy groceries  
2. [ ] Finish homework  
  
Choose an option: 3  
Enter task number to mark complete: 1  
Task marked as done.

# Possible Improvements

* Add due dates and priorities.
* Save data in a JSON or CSV file.
* Create a GUI version using Tkinter or a web version with Flask.
* Add search or filter (e.g., show only incomplete tasks).

# Notes

This is meant as a practice project. It can be expanded in many directions depending on what you want to learn:  
- File handling (saving tasks).  
- OOP design (Task class with attributes).  
- User interface (console vs. GUI vs. web).