To-Do List Program Documentation

# Introduction

This document provides detailed documentation for a Python-based to-do list application. The program allows users to manage tasks by adding, completing, organizing, and removing them, as well as updating task priorities and colors. The list is organized and provides feedback to the user for every action.

# Code Overview

## 1. Initializing Task List

The program initializes an empty list named `tasks` to store the user's tasks. Each task is represented as a dictionary that contains information such as task name, completion status, priority, and color.

## 2. Display Tasks

The function `display\_tasks()` is responsible for showing the list of tasks. It checks whether the task list is empty and provides feedback if no tasks are available. For each task, it displays the task name, priority, completion status, and associated color.

## 3. Adding a Task

The function `add\_task(task\_name, priority, color)` is used to add a new task to the list. It takes three arguments: the name of the task, its priority, and the color associated with it. The task is added to the `tasks` list with an initial completion status set to `False`.

## 4. Mark a Task as Completed

The function `mark\_completed(task\_number)` marks a specific task as completed by changing its `completed` status to `True`. The function takes the task number as input, validates it, and updates the task's status.

## 5. Remove a Task

The function `remove\_task(task\_number)` removes a task from the list. It takes the task number as input, validates the input, and removes the corresponding task from the `tasks` list.

## 6. Organizing Tasks by Priority

The function `organize\_by\_priority()` organizes tasks in the list by their priority. It sorts the tasks in ascending order of priority using the `sort()` function.

## 7. Update Task Color

The function `update\_task\_color(task\_number, color)` allows the user to update the color associated with a task. It takes the task number and the new color as input, validates the input, and updates the task's color.

## 8. Main Function

The `main()` function is the entry point of the program. It runs an infinite loop, displaying a menu of options to the user. The user can select an option to perform various operations like displaying tasks, adding tasks, marking them as completed, removing tasks, organizing tasks by priority, or updating the task color. The program will continue until the user selects the 'Quit' option.

# Conclusion

This to-do list program provides a simple yet functional way to manage tasks. The use of a list of dictionaries allows for the flexibility of adding multiple properties to each task, such as priority, color, and completion status. With an intuitive menu system, the user can easily navigate through the different functionalities and organize their tasks efficiently.