Countdown Timer Application (Tkinter)

Overview

This Python application provides a user-friendly countdown timer built with the Tkinter GUI framework. It allows users to:
- Enter time in either MM:SS format or plain seconds
- Start, stop, and reset the countdown
- Get notified when the timer ends (via popup and system bell)
- Quickly add predefined durations (+30s, +1m, +5m)
- Clear the timer with a single click

Features

- Input Flexibility: Accepts countdown input in MM:SS or just seconds.
- Live Countdown: Dynamically updates the timer display.
- Control Buttons:
  - Start: Begins the countdown.
  - Stop: Pauses the timer.
  - Reset: Resets the timer to the original input.
- Quick Add Buttons: Quickly add 30s, 1 minute, or 5 minutes.
- Clear Function: Clears input and display.
- Notification: Alert box and audible bell when time is up.
- Multithreaded: Timer runs in a separate thread for smooth UI responsiveness.

Setup Instructions

1. Requirements
- Python 3.x (tested with 3.7+)
- Standard library only - no external dependencies needed

2. Running the App
Option A: Run with Python
Save the code in a file named (e.g.) countdown_timer.py, then run:
python countdown_timer.py

Option B: From IDE
Open the file in any Python IDE (e.g., VS Code, PyCharm) and run the script.

Usage Guide

1. Input Time
- Type a time like 1:30 for 1 minute 30 seconds, or just 90 for 90 seconds.

2. Use Quick Add Buttons (Optional)
- Click +30s, +1m, or +5m to add time to your input.

3. Start the Timer
- Click Start. The countdown begins.

Countdown Timer Application (Tkinter)

4. Stop the Timer
- Click Stop to pause the timer.

5. Reset the Timer
- Click Reset to go back to the original time.

6. Clear All
- Click Clear to clear both the entry and timer display.

7. Notification
- When the countdown reaches zero, a popup message and bell sound will alert you.

How It Works

- The main timer logic runs in a background thread using Pythons threading.Thread to keep the UI responsive.
- Time parsing accepts both MM:SS and seconds using string manipulation and int() conversion.
- UI elements are built using Tkinter's layout managers (pack()).
- Error handling includes:
  - Invalid format (non-numeric, incorrect separators)
  - Negative or zero durations

Notes

- GUI dimensions are set to 380x320, but you can adjust them if needed.
- You can modify the self.root.bell() line to play custom sounds for more advanced alerts (using playsound or winsound module).

Optional Enhancements

Consider adding:
- Sound alert using a custom audio file
- Pause/Resume functionality
- Countdown history
- Dark mode or theme switcher
- Minimize to tray / system notifications