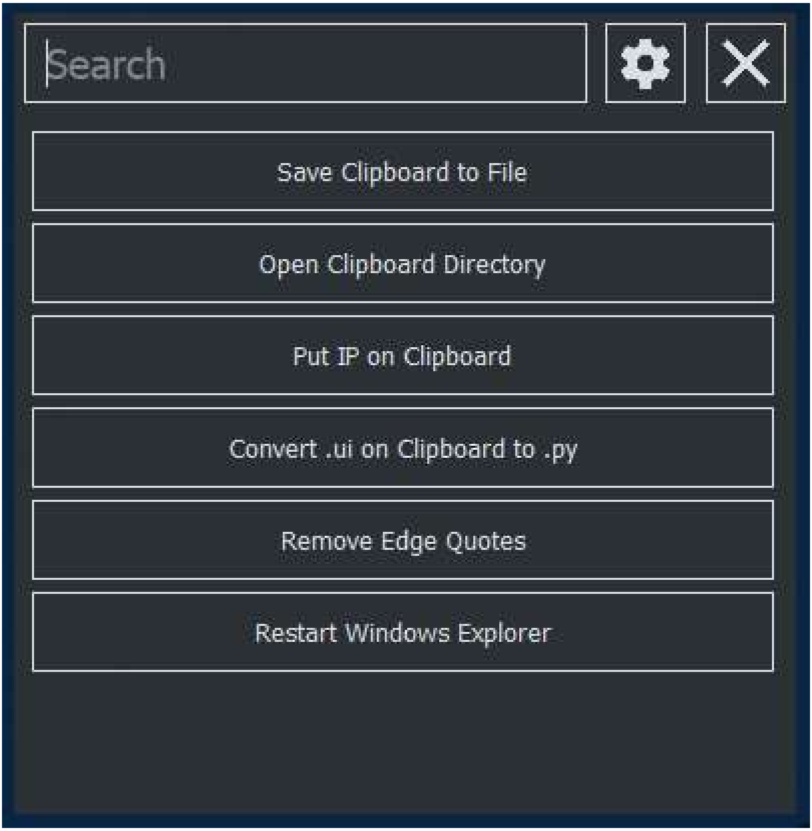
Quick Script

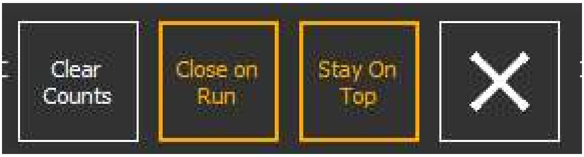
A small GUI that displays home-made scripts to be executed. The scripts will appear in a scrolling window and on selection they will run and the window will then close (can be configured). Sample scripts included.

What is this?

This project was my attempt to stop throwing out useful scripts that I made to make small tasks easier. I had always found myself making things and deleting them later as I didn't think I was going to use them again. This project allows you to attach a script easily to this project which makes things searchable and easy to run. I connected a hotkey to this to run smaller things.

Demonstration and Screenshots





Installation and Setup

1. Clone or download the git repo at https://github.com/brentvollebregt/quick-script
2. Install Requirements
   - Python 3
   - PYQT5 (pip install pyqt5)
3. Run quick-script.py / RunQuickScript.vbs (could attach it to a hotkey for easy access)

Usage

To run this the script quick-script.py needs to be run with Python. A window will appear with your scripts.

Configuration (settings.json)

This file is located by quick-script.py and keeps a record of settings and script run counts.

```json
{
    "close_on_run": true,
    "run_count": {},
    "stay_on_top": true,
    "window_height": 400
}
```

Python script files

To make a script appear in the UI they need to be in the /scripts/ folder. The script needs four main features; name, description, tags and a main method which returns True on success. An example of a script is shown here:

```python
NAME = "Example Script"
DESCRIPTION = "Run example script"
TAGS = ["example", "script"]

def main():
    print ("This is an example script")
    return True
```

The scripts can be named anything as long as they are in /scripts/ with a .py extension. The script does not have to have name, description or tags, they will just be rendered as none.

A parameter can be added to the main() method and as long as there is only one parameter required, the window class will be passed to main() so the method will now have access to the window and can use functions like dialogCritical().

Scripts Included

- Remove edge quotes: Will remove edge quotes from the string in your current clipboard (one or both sides)
- Save Clipboard to File: Will request where to save the current clipboard contents to. Supports images and text (will save to .bmp/.txt)
- Put IP on Clipboard: Will set your local IP to the clipboard.
- Restart Windows Explorer: Restarts windows explorer.
- UI to PY from Clipboard: Gets the current .ui file in the clipboard and will convert it to py using pyuic5 and stores it where the .ui file is located.
- Open Clipboard Directory: Open the location in the clipboard in windows explorer.
- Type Clipboard: Type clipboard contents at a slow pace.