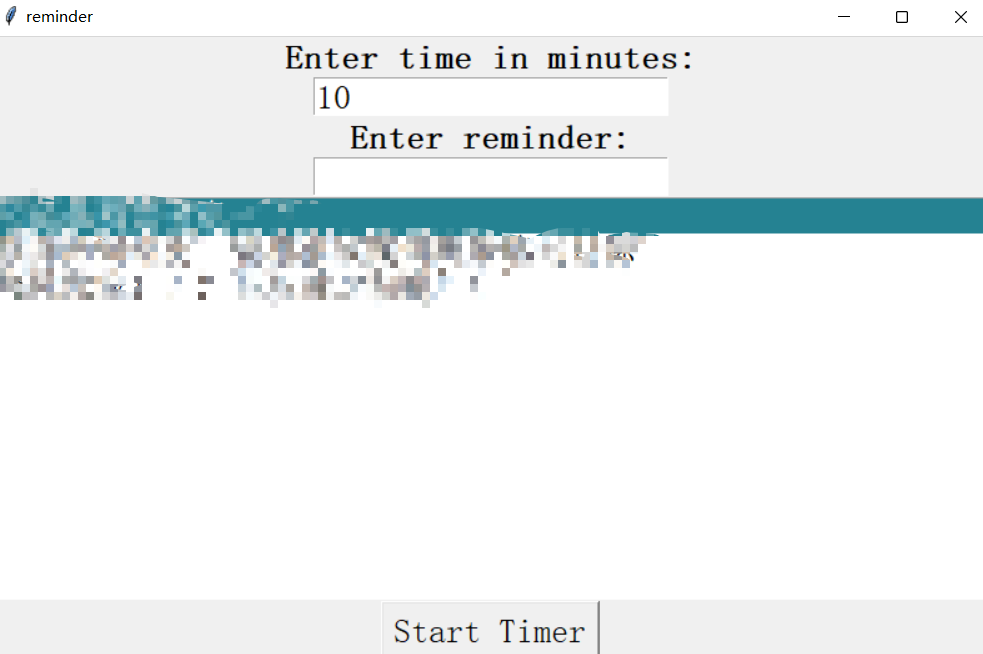
基于python的监控程序，下文以“monitor”指代

**一、监控前初始设定**

对于UI外观上没有要求，只要能实现功能即可（推荐使用Tkinter，也可以用其他简单的库）

开始运行之后，程序会一直在后台挂着（**所以要求程序涉及的库等要尽量体积小和少，减少内存占用**）

弹出窗口，设定运行时段。



类似于图中label+entry+button就ok

监控开始、结束有两种方式：①设定起始、中止时间时自动开始、中止监控。②手动选择立刻开始、结束（“立刻开始”按钮就在初始设定界面上，“立刻结束”选项则参考第三部分）

**二、监控过程——记录用户在这段时间里于什么时间打开了什么程序**

需要记录的数据说明：

使用时段之内所运行的应用程序的名称+使用时间（何时开启+何时关闭 | 如果是monitor监控开始之前就运行/结束后仍在运行 则以监控时段起始点/终点计 ）

记录的精度说明：

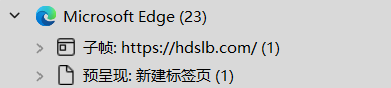
①应用程序的名称部分：

如果是浏览器，则需要精确到网址

示例：



实际打开的网址

任务管理器中的记录效果

最好能模仿任务管理器，输出程序“Microsoft Edge”+具体标签页的名称“罗斯塔夫来到了沙漠……”

需要注意的是，此时浏览器打开了两个标签页，monitor需要分别记录两个标签页的名称和打开时间。

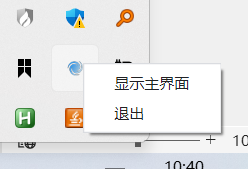
注：只需要记录应用程序即可，后台进程无需记录

②时间部分：精确到分钟即可（示例—— 15:45—16：30）

**三、监控结束后的数据输出**

达到先前设定的结束时间后，弹出弹窗询问用户是否继续。如果选择是，则立刻结束监控；反之，则继续监控，由用户之后手动选择结束。

手动结束：



以这种形式，右键后台图标，弹出“中止监控”选项即可

监控结束后，生成数据文件（命名最好含当日日期）

数据分为三列，依次为 程序名称 程序开始时间 程序结束时间

数据按照程序开始时间的先后排列

文件格式可以是txt、csv、excel等，以涉及的库小、占的内存少为选择依据。且最好格式上便于后续转成excel等进行统计分析（本程序不涉及统计分析部分）。