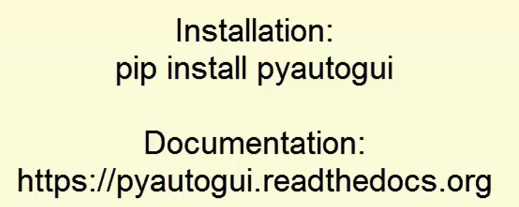
**GUI Automation**

**Controlling the Mouse from Python:**

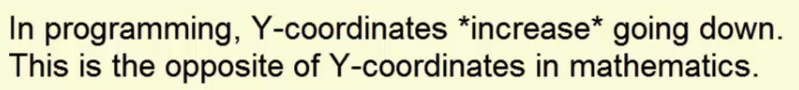


**Notes:**

Just think like your screen is a co-ordinate system with (x,y) co-ordinates.

The X is left and right of the screen. And Y is Up and Down part of the screen.

(0,0) is the top left of the screen.



>>> import pyautogui

>>> pyautogui.size() ### this will give the complete screen size

Size(width=1680, height=1050)

>>> width, height = pyautogui.size()

>>> width

1680

>>> height

1050

>>>

**How to get current position of mouse ?**

>>> pyautogui.position()

Point(x=1172, y=478)

>>> pyautogui.position()

Point(x=0, y=0)

>>> pyautogui.position()

Point(x=1679, y=1049)

**How to move the mouse cursor to an absolute position?**

>>> pyautogui.moveTo(10,10) ### It will move the mouse cursor to (10,10) position of screen

>>> pyautogui.moveTo(10,10, duration=1.5) # it will move the cursor but in 1.5 seconds like a human 😊

How to move the mouse cursor to relative position?

>>> pyautogui.moveRel(200,0)

>>> pyautogui.moveRel(0,200, duration=2)

>>> pyautogui.moveRel(0,-100)

>>> pyautogui.moveRel(0,100,duration=2)

>>> pyautogui.moveRel(0,100,duration=1)

>>> pyautogui.moveRel(0,-100,duration=2)

How to Click on some particular location?

>>> pyautogui.position()

Point(x=322, y=1039)

>>> pyautogui.click(322,1039)

>>> pyautogui.doubleClick(322,1039)

>>> pyautogui.rightClick(322,1039)

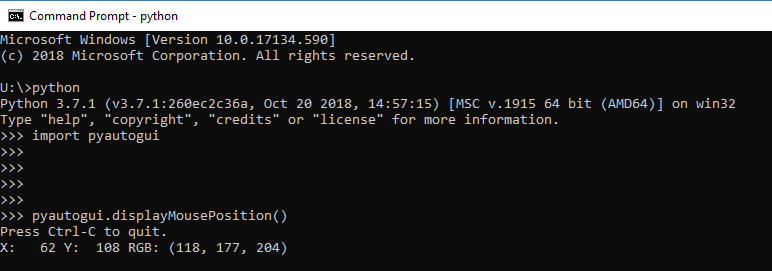
>>> pyautogui.middleClick(322,1039)

WAP to draw a spiral design on paint.

|  |
| --- |
| import pyautogui  pyautogui.click() #click to put drawing program in focus  distance = 200  while distance > 0:  print(distance, 0)  pyautogui.dragRel(distance,0, duration=1) #move right  distance=distance-25  print(0, distance)  pyautogui.dragRel(0, distance, duration=1) #move down  print(-distance, 0)  pyautogui.dragRel(-distance,0, duration=1) #move left  distance=distance-25  print(0, -distance)  pyautogui.dragRel(0, -distance, duration=1) #move up |
| O/p- |

How to get co-ordinates of mouse pointer on the screen with Command prompt?

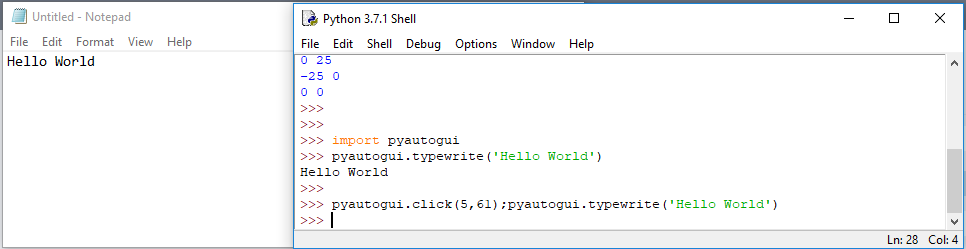
* With below method we can move the mose pointer and get the co-ordinates…



Notes:

1. Controlling the mouse and keyboard is called GUI automation.
2. The PyAutoGUI third-party module has many functions to control the mouse and keyboard.
3. pyautogui.size() returns the screen resolution, pyautogui.position() returns the mouse position, both as tuple of two ints.
4. pyautogui.moveTo() moves the mouse to an x, y coordinate.
5. The mouse move is instant, unless you pass an int for the duration keyword argument.
6. pyautogui.moveRel() moves the mouse relative to its current position.
7. pyautogui’s click(), doubleClick(), rightClick, and middleClick() click the mouse buttons.
8. dragTo() and deagRel() will move the mouse while holding down a mouse button.
9. If your program gets out of control, quickly move the mouse cursor to the top-left.
10. There’s more documentation at pyautogui.readthedocs.org

**Controlling the Keyboard from Python:**



>>> import pyautogui

>>> pyautogui.typewrite('Hello World')

Hello World

>>>

>>> pyautogui.click(5,61); pyautogui.typewrite('Hello World')

>>> pyautogui.click(5,61);pyautogui.typewrite('Hello World', interval=0.2) #interval is time between type each characters

>>> pyautogui.click(5,61);pyautogui.typewrite(['a','b','left','left','X','Y'], interval=0.2)

>>>

O/P-



We can use list of different keyboard keys.. please find the list of keys as below…

>>> pyautogui.KEYBOARD\_KEYS

['\t', '\n', '\r', ' ', '!', '"', '#', '$', '%', '&', "'", '(', ')', '\*', '+', ',', '-', '.', '/', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ':', ';', '<', '=', '>', '?', '@', '[', '\\', ']', '^', '\_', '`', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '{', '|', '}', '~', 'accept', 'add', 'alt', 'altleft', 'altright', 'apps', 'backspace', 'browserback', 'browserfavorites', 'browserforward', 'browserhome', 'browserrefresh', 'browsersearch', 'browserstop', 'capslock', 'clear', 'convert', 'ctrl', 'ctrlleft', 'ctrlright', 'decimal', 'del', 'delete', 'divide', 'down', 'end', 'enter', 'esc', 'escape', 'execute', 'f1', 'f10', 'f11', 'f12', 'f13', 'f14', 'f15', 'f16', 'f17', 'f18', 'f19', 'f2', 'f20', 'f21', 'f22', 'f23', 'f24', 'f3', 'f4', 'f5', 'f6', 'f7', 'f8', 'f9', 'final', 'fn', 'hanguel', 'hangul', 'hanja', 'help', 'home', 'insert', 'junja', 'kana', 'kanji', 'launchapp1', 'launchapp2', 'launchmail', 'launchmediaselect', 'left', 'modechange', 'multiply', 'nexttrack', 'nonconvert', 'num0', 'num1', 'num2', 'num3', 'num4', 'num5', 'num6', 'num7', 'num8', 'num9', 'numlock', 'pagedown', 'pageup', 'pause', 'pgdn', 'pgup', 'playpause', 'prevtrack', 'print', 'printscreen', 'prntscrn', 'prtsc', 'prtscr', 'return', 'right', 'scrolllock', 'select', 'separator', 'shift', 'shiftleft', 'shiftright', 'sleep', 'space', 'stop', 'subtract', 'tab', 'up', 'volumedown', 'volumemute', 'volumeup', 'win', 'winleft', 'winright', 'yen', 'command', 'option', 'optionleft', 'optionright']

>>>

>>> pyautogui.press('f1') #press a single key from keyboard

>>> pyautogui.hotkey('ctrl','o') # combination of two keys

Notes:

1. PyAutoGUI’s virtual key presses will go the window that currently has focus.
2. **typewrite()** can be passed a string characters to type. It also has a interval keyword argument.
3. Passing a list of strings to **typewrite()** lets you use hard-to-type keyboard keys, like ‘shift’ or ‘f1’.
4. These keyboard key strings are in the **pyautogui.KEYBOARD\_KEYS** list.
5. **pyautogui.press()** will press a single key.
6. **pyautogui.hotkey()** can be used for keyboard shortcuts, like **Ctrl+O**.

**Screenshots and Image Recognition:**

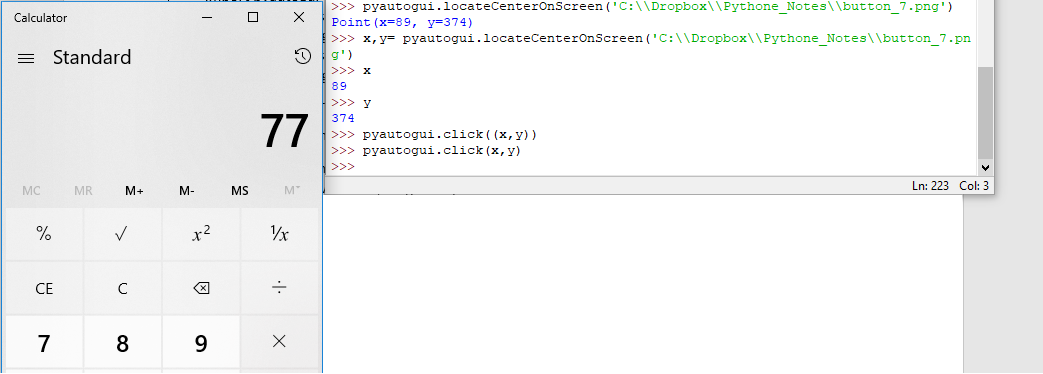
**WAP Take a screenshot…**

|  |
| --- |
| >>> import pyautogui  >>> pyautogui.screenshot()  <PIL.Image.Image image mode=RGB size=1680x1050 at 0x22B3A97FD68>  >>> pyautogui.screenshot('C:\\Dropbox\\Pythone\_Notes\\screenshot\_example.png')  <PIL.Image.Image image mode=RGB size=1680x1050 at 0x22B3AFB5160>  >>> |

**WAP to recognize the image on screen?**

>>> pyautogui.locateOnScreen('C:\\Dropbox\\Pythone\_Notes\\button\_7.png')

Box(left=53, top=350, width=72, height=48)



|  |
| --- |
| >>> pyautogui.locateCenterOnScreen('C:\\Dropbox\\Pythone\_Notes\\button\_7.png')  Point(x=89, y=374)  >>> x,y= pyautogui.locateCenterOnScreen('C:\\Dropbox\\Pythone\_Notes\\button\_7.png')  >>> x  89  >>> y  374  >>> pyautogui.click((x,y))  >>> pyautogui.click(x,y) |

Notes:

1. A screeshot is an image of the screen’s content.
2. The pyautogui.screenshot() will return an Image object of the screen, or you can pass it a filename to save it to a file.
3. locateOnScreen() is passed a simple image file and returns the co-ordinates of where it is on the screen.
4. locateCenterOnScreen() will return an (x,y) tuple of where the image is on the screen.
5. Combining the keyboard/mouse/screenshot functions let’s you make awesome stuff ! 😊