**current screen resolution width and height:**

>>> pyautogui.size(1920, 1080)

**Fail-Safes**

Set up a 2.5 second pause after each PyAutoGUI call:

>>> import pyautogui

>>> pyautogui.PAUSE = 2.5

**All clicks can be done with click(), but these functions exist for readability. Keyword args are optional:**

>>> pyautogui.rightClick(x=moveToX, y=moveToY)

>>> pyautogui.middleClick(x=moveToX, y=moveToY)

>>> pyautogui.doubleClick(x=moveToX, y=moveToY)

>>> pyautogui.tripleClick(x=moveToX, y=moveToY)

**Positive scrolling will scroll up, negative scrolling will scroll down:**

>>> pyautogui.scroll(amount\_to\_scroll, x=moveToX, y=moveToY)

>>> pyautogui.scroll(10, x=100, y=100) # move mouse cursor to 100, 200, then scroll up 10 "clicks"

**The full list of key names is in pyautogui.KEYBOARD\_KEYS.**

Keyboard hotkeys like Ctrl-S or Ctrl-Shift-1 can be done by passing a list of key names to hotkey():

>>> pyautogui.hotkey('ctrl', 'c') # ctrl-c to copy

>>> pyautogui.hotkey('ctrl', 'v') # ctrl-v to paste

**Message Box Functions**

If you need to pause the program until the user clicks OK on something, or want to display some information to the user, the message box functions have similar names that JavaScript has:

>>> pyautogui.alert('This displays some text with an OK button.')

>>> pyautogui.confirm('This displays text and has an OK and Cancel button.')

'OK'

>>> pyautogui.prompt('This lets the user type in a string and press OK.')

'This is what I typed in.'

**To press these keys, call the press() function and pass it a string from the pyautogui.KEYBOARD\_KEYS such as enter, esc, f1. See KEYBOARD\_KEYS.**

For example:

>>> pyautogui.press('enter') # press the Enter key

>>> pyautogui.press('f1') # press the F1 key

>>> pyautogui.press('left') # press the left arrow key

**The press() function is really just a wrapper for the keyDown() and keyUp() functions, which simulate pressing a key down and then releasing it up. These functions can be called by themselves. For example, to press the left arrow key three times while holding down the Shift key, call the following:**

>>> pyautogui.keyDown('shift') # hold down the shift key

>>> pyautogui.press('left') # press the left arrow key

>>> pyautogui.press('left') # press the left arrow key

>>> pyautogui.press('left') # press the left arrow key

>>> pyautogui.keyUp('shift') # release the shift key

**To press multiple keys similar to what typewrite() does, pass a list of strings to press(). For example:**

>>> pyautogui.press(['left', 'left', 'left'])

**The hotkey() Function**

To make pressing hotkeys or keyboard shortcuts convenient, the hotkey() can be passed several key strings which will be pressed down in order, and then released in reverse order. This code:

>>> pyautogui.hotkey('ctrl', 'shift', 'esc')

**KEYBOARD\_KEYS**

**The following are the valid strings to pass to the press(), keyDown(), keyUp(), and hotkey() functions:**

['**\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']

**The alert() Function**

>>> alert(text='', title='', button='OK')

Displays a simple message box with text and a single OK button. Returns the text of the button clicked on.

**The confirm() Function**

>>> confirm(text='', title='', buttons=['OK', 'Cancel'])

Displays a message box with OK and Cancel buttons. Number and text of buttons can be customized. Returns the text of the button clicked on.

**The prompt() Function**

>>> prompt(text='', title='' , default='')

Displays a message box with text input, and OK & Cancel buttons. Returns the text entered, or None if Cancel was clicked.