#### PyAutoGUI

更新日：1/26

# PyAutoGUIについて

PyAutoGUIは，マウスやキーボード操作を制御するためのPythonのモジュールである．GUIとは，キー入力やマウスを使ったコンピューターの操作手法を指し，PyAutoGUIでは，私たちが使っているGUI操作をコンピューターに自動で行うことができる．具体的にPyAutoGUIで出来ることは次の通りである．

|  |
| --- |
| 【PyAutoGUIで可能な操作】   * マウスの操作(移動/ボタン/スクロール) * キーボードの操作(テキスト入力/キー操作) * アラートウインドウの制御(通常のアプリケーションウインドウは操作無理) * イメージマッチング(bmp/jpg/pngで設定可能)   （※opencv\_python , pillow , Imageモジュールがあるとあいまい検索が可能）   * ピクセルカラー取得(特定座標または範囲の色を取得) * スクリーンショットの取得(全域/範囲) |

PyAutoGUIは外部ライブラリーなので、プログラム実行前にインストールが必要となる．インストールは，他のライブラリーと同様である．

# RPAツールにおけるPyAutoGUI

RPAツール内でPyAutoGUIの重要な操作は次の通り．

* マウスの操作(移動/ボタン/スクロール)
* キーボードの操作(テキスト入力/キー操作)

## マウスを操作する関数

マウスを操作する関数は次のとおりである．

表 1　マウスを操作する関数

|  |  |
| --- | --- |
| **関数** | **操作内容** |
| pyautogui.moveTo(x,y) | マウス座標を(x,y)まで移動 |
| pyautogui.moveRel(x,y) | マウス座標を(x,y)だけ移動する |
| pyautogui.position() | マウスの現在位置(x,y)を取得 |
| pyautogui.click() | マウスボタンをクリックする |
| pyautogui.mouseDown() | マウスボタンを押したままにする |
| pyautogui.mouseUp() | マウスボタンを離す |

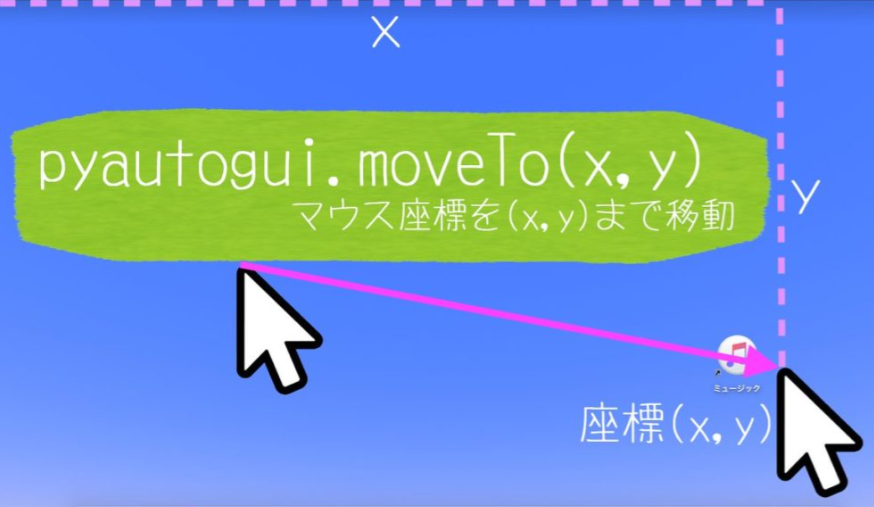
ちなみに座標は，モニタの左上をx=0, y=0として右下に行くに従って値が増加する．



図表 2‑1　モニタの座標の設定について

【マウスを指定した座標まで移動】

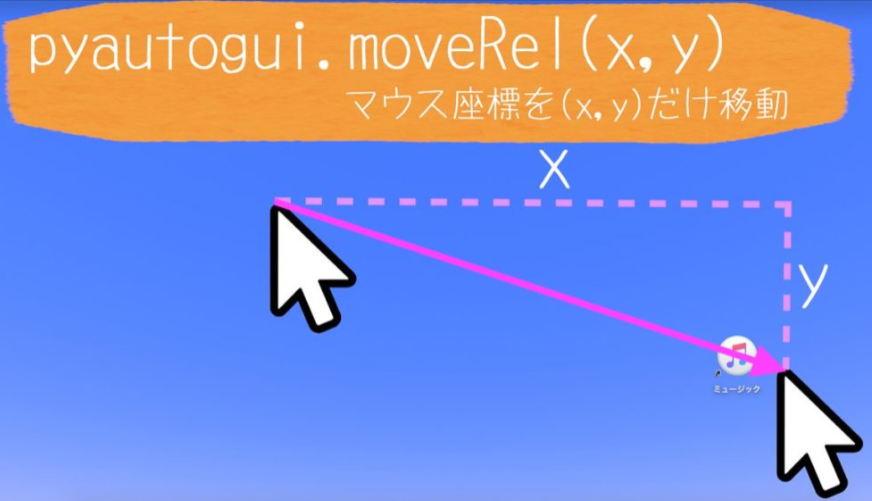
pyautogui.moveTo(x,y)で，マウスを指定した座標(x,y)まで移動させる．元のマウス座標に関係なく，引数の座標(x,y)までマウスを移動する．



図表 2‑2　マウス移動のイメージ①

【マウスを指定した座標だけ移動】

pyautogui.moveRel(x,y)で，マウスを指定した座標分(x,y)だけ移動させる．元のマウス座標から，引数の座標分(x,y)だけマウスを移動する．



図表 2‑3　マウス移動のイメージ②

【マウスの現在の座標を調べる】

pyautogui.position()で，マウスの現在の座標(x,y)を取得する．実行した時点のマウス座標を取得して，返り値(x,y)を返してくれる．



図表 2‑4　マウスの現在位置の取得イメージ

【マウスを指定したボタン・回数でクリック】

pyautogui.click()では，引数でクリックするボタン・回数も設定できる．

* x=クリックしたい位置のx座標
* y=クリックしたい位置のy座標
* button=クリックしたいボタン(left/right/middle)
* clicks=クリックする回数
* interval=クリックする間隔の秒数
* pause=クリック後に一時停止する秒数

また，クリック操作の関数も用意されているので、そちらでも簡単に実行することが出来る．

表 2　クリック操作の関数

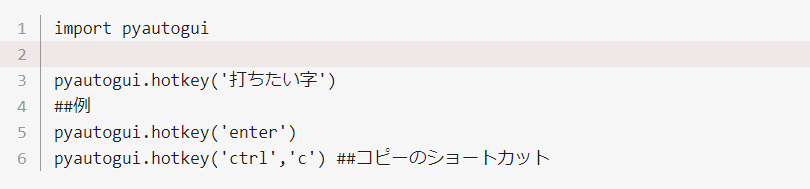
|  |  |
| --- | --- |
| **関数** | **操作内容** |
| pyautogui.click() | マウスを左クリック |
| pyautogui.rightClick() | マウスを右クリック |
| pyautogui.middleClick() | マウス中央ボタンをクリック |
| pyautogui.doubleClick() | マウスをダブルクリック |
| pyautogui.tripleClick() | マウスをトリプルクリック |

## キーボードを操作する関数

キーボードを操作する関数は次のとおりである．

表 3　キーボードを操作する関数

|  |  |
| --- | --- |
| **関数** | **操作内容** |
| pyautogui.press() | キーボードの()を押す |
| pyautogui.typewrite() | キーボードの()を順番にタイピング |
| pyautogui.hotkey() | キーボードの()を同時押しする |
| pyautogui.keyDown() | キーボードの()を押したままにする |
| pyautogui.keyUp() | キーボードの()を離す |



図表 2‑5　キーボードを操作する関数の使用例

【キーボードを入力してタイピング】

pyautogui.press(‘キー’)は、キーボード上の指定したキーを1つ押す関数である．pyautogui.hotkey(‘キー1′,’キー2’)では、キーボードの複数のキーを同時押しできる．pyautogui.typewrite(‘文字列’)なら，1文字以上の文字列を入力できる．

表 4　キーボードの入力

|  |  |
| --- | --- |
| **キーボード** | **入力** |
| Enterキー | ‘enter’,’retuen’,’\n’ |
| Escキー | ‘esc’ |
| Shiftキー | ‘shiftleft’,’shiftright’ |
| Altキー | ‘altleft’,’altright’ |
| Ctrlキー | ‘ctrlleft’,’ctrlright’ |
| Tabキー | ‘tab’,’\t’ |
| Backspaceキー・Deleteキー | ‘backspace’,’delete’ |
| PageUpキー・PageDownキー | ‘pageup’,’pagedown’ |
| Homeキー・Endキー | ‘Home’,’end’ |
| 矢印キー(↑↓←→) | ‘up’,’down’,’left’,’right’ |
| ファンクションキー | ‘f1′,’f2’,’f3’など |
| 音量コントロールキー | ‘volumeup’,’volumedown’,’volumemute’ |
| Pauseキー | ‘pause’ |
| CapsLockキー | ‘capslock’ |
| NumLockキー | ‘numlock’ |
| ScrollLockキー | ‘scrolllock’ |
| Insキー | ‘insert’ |
| PrintScreenキー | ‘printscreen’ |
| Winキー(Windowsのみ) | ‘winleft’,’winright’ |
| Commandキー(Macのみ) | ‘command’ |
| Optionキー(Macのみ) | ‘option’ |

pyautogui.typewrite()では、英数字しか入力できないので注意が必要である．日本語を自動入力するには，文字列をクリップボードにコピー&ペーストして行う．