

Leon

pyautogui

3.5 保护措施

```
.FAILSAFE=True
```

4 鼠标函数

4.1 鼠标移动

```
绝对坐标 .moveTo(x,y,duration=1)  
相对坐标 .moveRel()
```

4.2 鼠标点击

```
.click(50,100,clicks=3,interval=0.25,button='right')  
.rightClick()  
.middleClick()  
.doubleClick()
```

4.3 鼠标滚轮

```
.hscroll()水平滚动  
.vscroll()垂直滚动
```

4.4 按下和弹开

```
.mouseDown(x=100,y=200,button='left')  
.mouseUp(x=100,y=200,button='left')
```

4.5 拖拽

```
.dragTo(100,200,2,button='right')  
.dragRel(30,0,2,button='left')
```

4.6 鼠标速度渐变

```
.moveTo(x,y,pyautogui.easeInQuad)  
pyautogui.easeInQuad  
pyautogui.easeOutQuad  
pyautogui.easeInOutQuad  
pyautogui.easeInBounce  
pyautogui.easeInElastic
```

5 键盘控制函数

5.1 字符输入

```
.typewrite("hello world\n",interval = 0.25)
```

5.2 模拟按键

```
press  
hotkey  
等价于  
*.keyDown()  
*.keyUp()  
*.keyDown('ctrl')  
*.keyDown('a')  
*.keyUp('ctrl')  
*.keyUp('a')
```

5.3 keyboard表

6 消息弹窗函数

6.1 单按钮消息弹窗

```
.alert(text=" ",title=" ",button='ok')
```

6.2 多按钮的消息弹窗

```
.confirm(text=" ",title=" ",button=range(10))
```

6.3 可以输入的消息弹窗

```
.prompt(text=" ",title=" ",default=" ")
```

6.4 可以输入密码的消息弹窗

```
.password(text=" ",title=" ",default=" ",mask=" ")
```

有返回值

7 截图函数

需要Pillow/PIL库

7.1 截图函数

```
im= *.screenshot ('im.png',region=(0,0,300,400))
```

7.2 识别定位

```
pyautogui.locateOnScreen('someButton.png', region=(0,0, 300, 400))
```

7.3 获取中心点

```
pyautogui.center((643, 745, 70, 29))
```

7.4 识别并返回中心点

```
pyautogui.locateCenterOnScreen('pyautogui/calc7key.png')
```

7.5 更多识别

```
locateOnScreen(image,grayscale=False) 返回image屏幕第一个找到的实例坐标  
locateAllOnScreen(image, grayscale=False) 返回一个生成器, 该生成器生成 (左, 顶部, 宽度, 高度) 元组, 用于在屏幕上找到图像的位置。
```

7.6 灰度匹配

```
pyautogui.locateOnScreen(' calc7key.png', grayscale=True)
```

8 像素匹配

8.1 获取坐标点像素值

```
im = pyautogui.screenshot() # 先获取屏幕截图  
im.getpixel((100, 200)) # 再获取截图x,y的RGB值  
等价于
```

8.2 pixel()获取像素值

```
pyautogui.pixel(100,200)
```

8.3 像素值检测

```
pyautogui.pixelMatchesColor(100, 200, (255, 255, 245), tolerance=10) 返回布尔值, tolerance为允许误差范围
```