

# Folha de Dicas do PyAutoGUI

Esta é uma referência rápida para usar o PyAutoGUI, um módulo de automação de interface gráfica multiplataforma que funciona em Python 2 e 3. Você pode controlar o mouse e o teclado, bem como realizar reconhecimento básico de imagens para automatizar tarefas no seu computador.

Todos os argumentos de palavras-chave nos exemplos são opcionais.

## Instalação

```
pip install pyautogui
```

## Funções Gerais

- `pyautogui.position()`: Retorna a posição atual do mouse (x, y).

```
(968, 56)
```

- `pyautogui.size()`: Retorna a resolução da tela (largura, altura).

```
(1920, 1080)
```

- `pyautogui.onScreen(x, y)`: Retorna `True` se as coordenadas x e y estiverem na tela.

```
True
```

## Dispositivos de Segurança

Configure uma pausa de 2,5 segundos após cada chamada do PyAutoGUI:

```
import pyautogui
pyautogui.PAUSE = 2.5
```

Quando o modo de segurança está ativado (`True`), mover o mouse para o canto superior esquerdo levanta uma `pyautogui.FailSafeException` que pode abortar o programa:

```
import pyautogui
pyautogui.FAILSAFE = True
```

## Funções do Mouse

As coordenadas XY têm origem (0, 0) no canto superior esquerdo da tela. X aumenta para a direita, Y aumenta para baixo.

- `pyautogui.moveTo(x, y, duration=num_seconds)`: Move o mouse para as coordenadas XY.
- `pyautogui.moveRel(xOffset, yOffset, duration=num_seconds)`: Move o mouse em relação à sua posição atual.

Se `duration` for 0 ou não especificado, o movimento é imediato. Observação: arrastar no Mac não pode ser imediato.

- `pyautogui.dragTo(x, y, duration=num_seconds)`: Arrasta o mouse para as coordenadas XY.
- `pyautogui.dragRel(xOffset, yOffset, duration=num_seconds)`: Arrasta o mouse em relação à sua posição atual.

A chamada `click()` consiste em clicar uma vez com o botão esquerdo na posição atual do mouse, mas argumentos podem mudar isso:

```
pyautogui.click(x=moveToX, y=moveToY, clicks=num_of_clicks, interval=secs_between_clicks, button='left')
```

O argumento `button` pode ser `'left'`, `'middle'` ou `'right'`.

- `pyautogui.rightClick(x=moveToX, y=moveToY)`
- `pyautogui.middleClick(x=moveToX, y=moveToY)`
- `pyautogui.doubleClick(x=moveToX, y=moveToY)`
- `pyautogui.tripleClick(x=moveToX, y=moveToY)`

A rolagem positiva rola para cima, a negativa para baixo:

```
pyautogui.scroll(amount_to_scroll, x=moveToX, y=moveToY)
```

Eventos individuais de botão:

```
pyautogui.mouseDown(x=moveToX, y=moveToY, button='left')
pyautogui.mouseUp(x=moveToX, y=moveToY, button='left')
```

## Funções do Teclado

Os pressionamentos de tecla vão para onde o cursor do teclado estiver.

```
pyautogui.typewrite('Hello world!\n', interval=secs_between_keys)
```

Uma lista de nomes de teclas também pode ser passada:

```
pyautogui.typewrite(['a', 'b', 'c', 'left', 'backspace', 'enter', 'f1'], interval=secs_between_keys)
```

Atalhos de teclado:

```
pyautogui.hotkey('ctrl', 'c') % ctrl-c to copy
pyautogui.hotkey('ctrl', 'v') % ctrl-v to paste
```

Eventos individuais de tecla:

```
pyautogui.keyDown(key_name)
pyautogui.keyUp(key_name)
```

## Funções da Caixa de Mensagens

```
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.'
```

A função `prompt()` retorna `None` se o usuário clicar em Cancelar.

## Funções de Captura de Tela

O PyAutoGUI usa Pillow/PIL para dados de imagem. No Linux, instale `scrot`:

```
sudo apt-get install scrot
```

```
pyautogui.screenshot() % returns a Pillow/PIL Image object
<PIL.Image.Image image mode=RGB size=1920x1080 at 0x24C3EF0>
pyautogui.screenshot('foo.png') % saves to a file
<PIL.Image.Image image mode=RGB size=1920x1080 at 0x31AA198>
```

Localizar imagem na tela:

```
pyautogui.locateOnScreen('looksLikeThis.png') % returns (left, top, width,
height)
(863, 417, 70, 13)
```

Localizar todos os locais da imagem:

```
list(pyautogui.locateAllOnScreen('looksLikeThis.png'))
[(863, 117, 70, 13), (623, 137, 70, 13), ...]
```

Localizar o centro da imagem:

```
pyautogui.locateCenterOnScreen('looksLikeThis.png') % returns center x and y
(898, 423)
```

As funções de localização retornam `None` se a imagem não for encontrada.