

# Resumo da Aula: Automação de Screenshots com PyAutoGUI

Esta aula foca na automação de capturas de tela (screenshots) utilizando o PyAutoGUI, um módulo Python para controle de interface gráfica. Abaixo, você encontra um resumo dos conceitos abordados, exemplos práticos e explicações dos códigos apresentados.

## Importação e Configuração Inicial

Para automatizar screenshots, importamos o módulo `pyautogui` e, opcionalmente, o módulo `time` para adicionar pausas entre comandos:

```
import pyautogui
import time
```

Configuramos pausas para facilitar a execução:

```
pyautogui.PAUSE = 1 % Pausa de 1 segundo entre comandos
pyautogui.FAILSAFE = True % Ativa o modo de segurança
```

## Captura de Tela Simples

O comando `pyautogui.screenshot()` captura a tela e pode salvar a imagem em um arquivo:

```
pyautogui.screenshot('exemplo.png')
```

Este código tira um screenshot e salva como `exemplo.png` na pasta atual.

## Automatizando Ações com Mouse e Screenshot

Um exemplo prático combina movimentos do mouse, cliques e capturas de tela. O script abaixo move o mouse para a posição (1751, 33), clica para minimizar uma janela e tira um screenshot:

```
import pyautogui
import time

pyautogui.PAUSE = 1
pyautogui.FAILSAFE = True

pyautogui.moveTo(1751, 33, duration=1) % Move o mouse em 1 segundo
time.sleep(1) % Pausa de 1 segundo
pyautogui.click() % Clica para minimizar
time.sleep(1) % Pausa de 1 segundo
pyautogui.screenshot('exemplo2.png') % Tira o screenshot
```

Este script minimiza uma janela (ex.: Visual Studio Code) e salva a captura como `exemplo2.png`.

## Monitoramento Contínuo com Screenshots

Para monitorar uma tela em tempo real, usamos um loop `while True` com `time.time()` para gerar nomes de arquivos únicos:

```
import pyautogui
import time

pyautogui.PAUSE = 1
pyautogui.FAILSAFE = True

while True:
    pyautogui.screenshot(f'print_{time.time()}.png') % Nome único com timestamp
    time.sleep(3) % Pausa de 3 segundos
```

Este código tira screenshots a cada 3 segundos, salvando cada imagem com um nome único baseado no horário (ex.: `print_1634567890.123456.png`). Para interromper, pressione `Ctrl+C`.

## Pontos-Chave

- **Facilidade:** Capturar screenshots com PyAutoGUI é simples, usando apenas `screenshot()`.
- **Automação:** Combine movimentos do mouse (`moveTo`, `click`) com screenshots para interagir com interfaces gráficas.
- **Monitoramento:** Use loops e `time.sleep()` para capturas automáticas em intervalos regulares.
- **Segurança:** Sempre ative `FAILSAFE` para interromper scripts movendo o mouse ao canto superior esquerdo.

## Dicas para Alunos

- Teste em um ambiente seguro para evitar cliques indesejados.
- Use `time.sleep()` para pausas, facilitando a depuração.
- Verifique a resolução da tela com `pyautogui.size()` para ajustar coordenadas.
- Para monitoramento, ajuste o intervalo (`time.sleep(30)` para 30 segundos, por exemplo).