

Como tirar screenshots com python

Capturar screenshots automaticamente com Python é uma habilidade incrível para desenvolvedores, profissionais de QA, analistas e até mesmo criadores de conteúdo. Se você está pensando em tirar capturas de tela para testes automatizados, monitorar sites, criar relatórios ou até mesmo gerar prévias de páginas da web, o Python tem ferramentas fantásticas para te ajudar.

Neste artigo, você vai descobrir como fazer isso usando as bibliotecas mais populares, com exemplos práticos e um guia passo a passo. Pronto para automatizar a captura de tela com apenas algumas linhas de código? Então, vamos nessa!

Por que usar Python para tirar screenshots?

Uma das características marcantes do Python é a facilidade com que se lê e escreve o código. Para completar, ele oferece diversas bibliotecas ótimas para automatizar tarefas, por exemplo, PyAutoGUI, Selenium e Playwright. Optar pelo Python para tirar prints de tela traz os seguintes benefícios:

Simple de programar e dar manutenção

Funciona para prints no próprio computador ou em outro

Perfeito para programas de computador, sites e celulares

Roda em Linux, Windows e macOS sem problemas

Conecta facilmente com APIs, bancos de dados e serviços online

Algumas ferramentas que você pode usar

1. pyautogui: Captura da tela inteira

Para capturar a tela do computador de alguém, usar pyautogui apresenta-se como uma solução bastante ágil e descomplicada.

`pip install pyautogui`

exemplo:

`import pyautogui`

`screenshot = pyautogui.screenshot()`

```
screenshot.save("tela.png")
```

Essa maneira de fazer as coisas é perfeita para automatizar ações na tela ou quando é preciso registrar tudo que aparece no monitor, até mesmo programas de computador.

2. Selenium: Capturando sites e páginas da web

Se você precisa tirar screenshots de páginas da web, o Selenium é uma das bibliotecas mais queridas no mundo do Python.

instalação:

```
pip install selenium
```

Você também vai precisar baixar o ChromeDriver compatível com sua versão do Google Chrome.

Exemplo:

```
from selenium import webdriver

driver = webdriver.Chrome()
driver.get("https://example.com")
driver.save_screenshot("screenshot.png")
driver.quit()
```

3. Playwright: Captura de sites com alta performance

Se você está em busca de desempenho e recursos modernos, o Playwright é a escolha perfeita. Ele é parecido com o Puppeteer, mas oferece suporte a vários navegadores.

Instalação:

```
pip install playwright
playwright install
```

Exemplo:

```
from playwright.sync_api import sync_playwright

with sync_playwright() as p:
    browser = p.chromium.launch()
    page = browser.new_page()
    page.goto("https://example.com")
    page.screenshot(path="screenshot.png", full_page=True)
    browser.close()
```

Esse método permite capturar a página inteira, bloquear anúncios e até exportar para PDF.

Vamos falar sobre algumas aplicações práticas:

Testes automatizados (QA): Capture o estado visual de uma página para garantir que as alterações no frontend estejam corretas.

Web scraping visual: Use junto com o BeautifulSoup para coletar e visualizar páginas da web.

Monitoramento de sites: Programe capturas de tela para acompanhar mudanças visuais em páginas específicas.

Geração de relatórios visuais: Adicione imagens em relatórios em HTML ou PDF.

Criação de prévias para apps de marketing: Capture automaticamente prévias de landing pages e campanhas.

Dicas e boas práticas:

Use `full_page=True` para garantir que você capture todo o conteúdo do site.

Combine com Pillow para editar, redimensionar ou cortar suas imagens.

Programe capturas automáticas usando `schedule`, `APScheduler` ou `cron`.

E lembre-se: evite capturar conteúdo sensível sem a devida permissão (fique atento à LGPD/GDPR).

Conclusão

Neste artigo, você descobriu como fazer screenshots com Python utilizando várias ferramentas para diferentes finalidades. Seja com `pyautogui`, `Selenium`, `Playwright` ou através de APIs, existe uma solução para cada necessidade — desde automação de desktop até a captura em larga escala de sites.

Caso você tenha interesse de fazer isso de uma maneira fácil, simples e escalável considere usar nossa api de screenshots!

