

Automatize Tudo

O Modelo Fordista com Python



Aprenda a visão de eficiência e inovação de Henry Ford com o poder da automação usando Python

MICHEL MONTALVAO

PRINCIPAIS AUTOMATIZADORES

Simplificando o dia a dia de suas tarefas

Os principais métodos para automatizar tarefas com Python envolvem a manipulação de arquivos, permitindo criar, renomear e organizar documentos; o envio automatizado de e-mails para notificações ou relatórios; a automação web com Selenium, que possibilita interagir com sites, como preencher formulários ou realizar logins automáticos; o processamento de dados com bibliotecas como pandas, ideal para manipular grandes volumes de informações em planilhas; o agendamento de tarefas usando schedule para execução automática em horários específicos; a integração com APIs para consumir serviços externos; e o gerenciamento de bancos de dados, que inclui a execução de consultas e atualizações automatizadas. Cada método simplifica tarefas repetitivas, economizando tempo e reduzindo erros manuais. Prepare-se para simplificar seu dia a dia!



01

Manipulação de Arquivos


Manipulação de Arquivos

A manipulação de arquivos em Python é utilizada quando você precisa trabalhar com dados que estão armazenados em arquivos, como ler, escrever, ou modificar o conteúdo desses arquivos. Isso é comum em várias situações, como:

1. **Leitura de arquivos:** Para carregar dados de arquivos para processamento no seu programa. Por exemplo, ler arquivos de texto (.txt), arquivos CSV, JSON, XML, entre outros.

Exemplo:

python


 Copiar código

```
with open('arquivo.txt', 'r') as file:  
    conteúdo = file.read()
```

2. **Escrita de arquivos:** Quando você precisa salvar ou registrar informações em um arquivo. Isso pode ser útil para gerar logs, exportar resultados, ou armazenar dados para uso posterior.

Exemplo:

python

 Copiar código

```
with open('novo_arquivo.txt', 'w') as file:  
    file.write('Olá, mundo!')
```

02

Automação de E-mails

AUTOMAÇÃO E-MAILS

A automação de e-mails em Python é utilizada quando você deseja enviar e-mails de forma programada ou em resposta a determinadas condições, sem precisar fazer isso manualmente. Algumas situações comuns em que a automação de e-mails pode ser útil incluem:

Exemplo 1: Enviar e-mail simples com `smtplib`

```
python

import smtplib
from email.mime.text import MIMEText

# Configuração do servidor e credenciais
smtp_server = 'smtp.gmail.com'
email_user = 'seu_email@gmail.com'
email_pass = 'sua_senha'

# Criar e-mail
msg = MIMEText('Conteúdo do e-mail')
msg['Subject'] = 'Assunto do E-mail'
msg['From'] = email_user
msg['To'] = 'destinatario@example.com'

# Enviar e-mail
with smtplib.SMTP(smtp_server, 587) as server:
    server.starttls()
    server.login(email_user, email_pass)
    server.sendmail(email_user, 'destinatario@example.com', msg.as_string())

print("E-mail enviado!")
```

[Copiar código](#)

03

Automação Web

AUTOMAÇÃO WEB

A automação web em Python é utilizada quando você precisa interagir automaticamente com websites, como preencher formulários, realizar login, navegar por páginas, coletar dados ou até mesmo realizar testes em sites. Algumas situações comuns em que a automação web é útil incluem:

1. **Selenium:** Permite interagir com um navegador de forma programática, simulando ações de um usuário, como clicar, digitar e navegar em páginas.

Exemplo básico de uso do Selenium:

```
python 📄 Copiar código  
  
from selenium import webdriver  
  
# Iniciar o navegador  
driver = webdriver.Chrome()  
  
# Abrir uma página  
driver.get('https://www.exemplo.com')  
  
# Interagir com a página  
search_box = driver.find_element('name', 'q')  
search_box.send_keys('automação web')  
search_box.submit()  
  
# Fechar o navegador  
driver.quit()
```



AGRADECIMENTOS



OBRIGADO POR LER ATÉ AQUI

Esse Ebook foi gerado por IA, e diagramado por humano.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizada uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.