

PENSE MENOS, PRODUZA MAIS!

PYTHON AUTOMAÇÃO SUPREMA

EDMUNDO
BATISTA



Python na Prática

Da Lógica à Automação Web

Python é uma linguagem poderosa, fácil de aprender e muito usada em automação, análise de dados e inteligência artificial.

Neste ebook, você vai conhecer os principais conceitos, comandos e bibliotecas, além de aprender como automatizar tarefas na web — tudo explicado de forma direta e prática.



01

CONCEITOS ESSENCIAIS

Antes de automatizar, é importante dominar o básico.

Conceitos Essenciais

Variáveis

Servem para guardar informações.

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

nome = "Edmundo"
idade = 30
print(f"Olá, {nome}! Você tem {idade} anos.")
```

Tipos de Dados

- **int**: números inteiros → 10
- **float**: números decimais → 3.14
- **str**: textos → "Python"
- **bool**: verdadeiro ou falso → True, False

Conceitos Essenciais

Estruturas de Controle

Usadas para decidir o fluxo do programa.

```
PYTHON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

nota = 8
if nota ≥ 7:
    print("Aprovado!")
else:
    print("Recuperação.")
```

Conceitos Essenciais

Laços de Repetição

Executam ações repetidamente.

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

for i in range(5):
    print("Executando...", i)
```

02

Funções: Tornando o Código Reutilizável

Funções permitem organizar o código em blocos reutilizáveis.

Funções

Tornando o Código Reutilizável

Funções permitem organizar o código em blocos reutilizáveis.

```
PYTHON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

def saudacao(nome):
    return f"Olá, {nome}! Seja bem-vindo."

print(saudacao("Pythonista"))
```

💡 **Dica:** Crie funções sempre que precisar repetir uma ação em vários pontos do seu código.

03

Principais Bibliotecas para Automação

Python tem milhares de bibliotecas prontas.

Principais Bibliotecas para Automação

Selenium

Controla o navegador automaticamente.

Instalação:

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

pip install selenium
```



Principais Bibliotecas para Automação

Selenium

Exemplo de login automático:

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

from selenium import webdriver
from selenium.webdriver.common.by import By
import time

driver = webdriver.Chrome()
driver.get("https://exemplo.com/login")

driver.find_element(By.ID, "usuario").send_keys("meu_usuario")
driver.find_element(By.ID, "senha").send_keys("minha_senha")
driver.find_element(By.ID, "botao_login").click()

time.sleep(3)
print("Login feito com sucesso!")
driver.quit()
```

Principais Bibliotecas para Automação

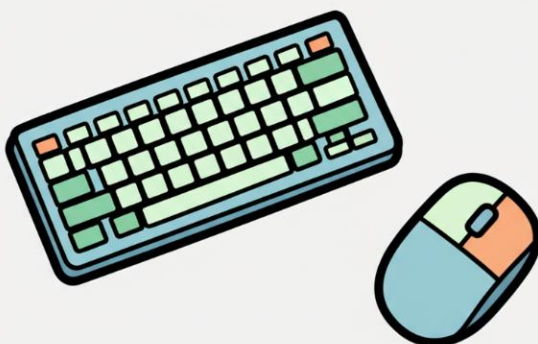
PyAutoGUI

Controla o mouse e o teclado para automatizar qualquer interface.

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY
```

```
pip install pyautogui
```

PyAutoGUI



Principais Bibliotecas para Automação

PyAutoGUI

Exemplo: abrir o menu iniciar e digitar algo.

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

import pyautogui
import time

pyautogui.press("win")
time.sleep(1)
pyautogui.write("Calculadora")
pyautogui.press("enter")
```


Principais Bibliotecas para Automação

Requests + BeautifulSoup

Perfeitos para coletar e preencher dados em sites sem precisar abrir o navegador.

Instalação:

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY
```

```
pip install requests beautifulsoup4
```



BeautifulSoup

Principais Bibliotecas para Automação

Requests + BeautifulSoup

Exemplo: coletar o título de um site.

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

import requests
from bs4 import BeautifulSoup

site = requests.get("https://www.python.org")
soup = BeautifulSoup(site.text, "html.parser")

titulo = soup.title.text
print("Título do site:", titulo)
```

04

Automatizando Preenchimento de Formulários

Preencher um formulário de cadastro automaticamente.

Automatizando Preenchimento de Formulários

Um exemplo real: preencher um formulário de cadastro automaticamente.

```
PYTHON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

from selenium import webdriver
from selenium.webdriver.common.by import By
import time

driver = webdriver.Chrome()
driver.get("https://exemplo.com/formulario")

driver.find_element(By.NAME, "nome").send_keys("João da Silva")
driver.find_element(By.NAME, "email").send_keys("joao@email.com")
driver.find_element(By.NAME, "mensagem").send_keys("Olá! Testando automação.")
driver.find_element(By.CSS_SELECTOR, "button.enviar").click()

time.sleep(2)
print("Formulário enviado!")
driver.quit()
```

05

Dicas Finais para Produtividade

Pequenas práticas que multiplicam seus resultados.

Dicas Finais para Produtividade

Truques simples para turbinar sua rotina com Python

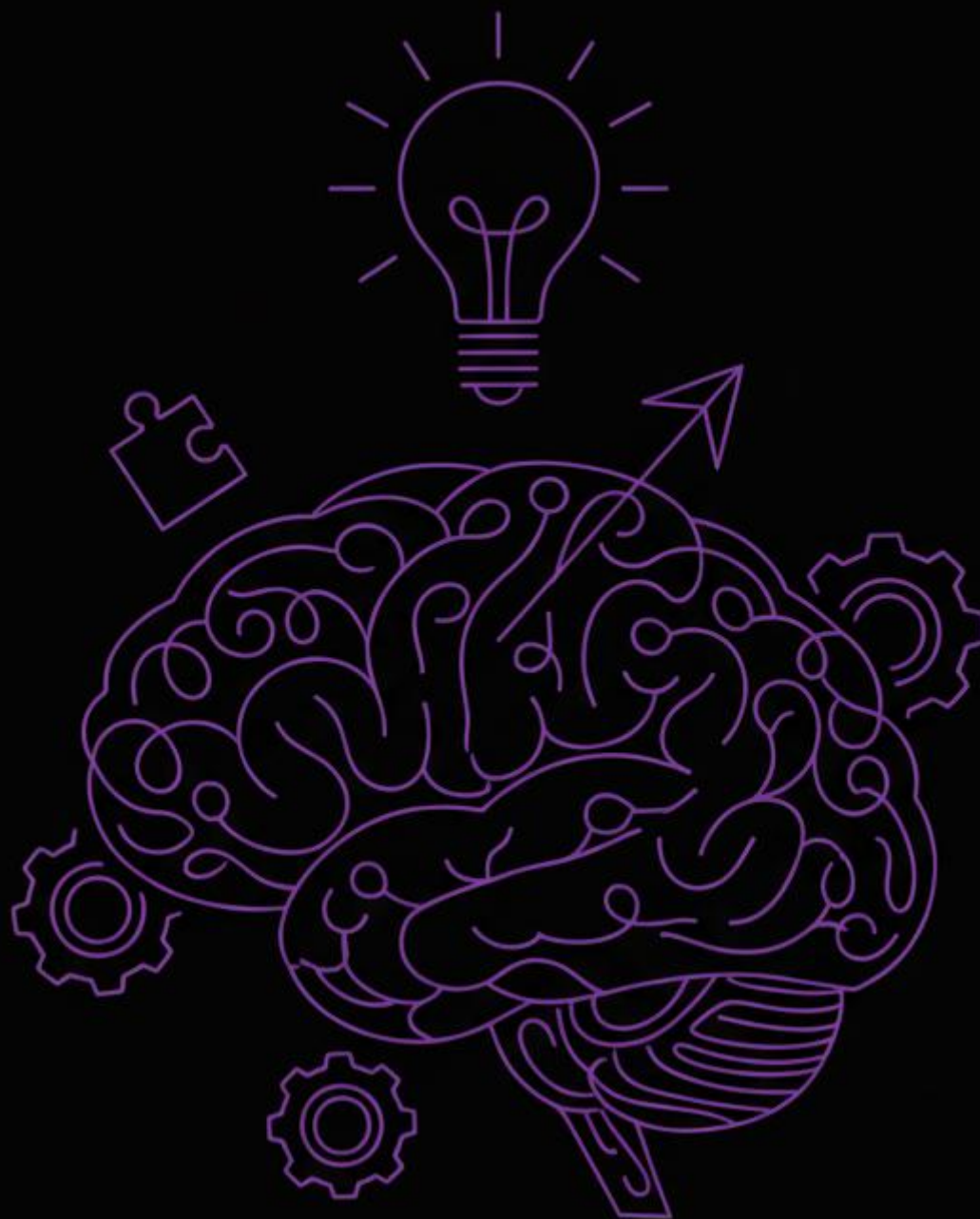
- ✓ **Use funções** para organizar suas automações.
- ✓ **Comente o código** para lembrar o que cada parte faz.
- ✓ **Crie scripts pequenos** e vá testando aos poucos.
- ✓ **Salve dados em CSV ou Excel** para integração com planilhas.

Exemplo:

```
PYHTON AUTOMAÇÃO SUPREMA - EDMUNDO BATISTA.PY

import pandas as pd

dados = {"Nome": ["Ana", "Bruno"], "Idade": [25, 30]}
df = pd.DataFrame(dados)
df.to_excel("pessoas.xlsx", index=False)
```



CONCLUSÃO E AGRADECIMENTOS

Agora é hora de colocar o código em ação!

Conclusão

Transforme conhecimento em automação real!

Com poucos comandos, você já é capaz de **automatizar tarefas repetitivas, coletar dados da web e preencher formulários automaticamente.**

O segredo é começar simples e evoluir aos poucos — Python faz o resto.



Obrigado pelo seu tempo dedicado!

Este material foi produzido com o suporte de ferramentas de inteligência artificial e diagramado manualmente.

O processo de criação está documentado no meu GitHub.

Por se tratar de um experimento educacional, o conteúdo não passou por validação humana rigorosa e pode apresentar inconsistências.



<https://github.com/eddgh/ebook-feito-com-ia.git>

 Autor:



EDMUNDO BATISTA



<http://www.linkedin.com/in/edmundo-josé-3660b76a>



<https://github.com/eddgh>