



RASTA PYTHON

A
Jornada do
<Codigo_Consciente>

Aprenda práticas de códigos sustentáveis

Lennon Müller

INTRODUÇÃO

Python e Sustentabilidade

Python é uma linguagem poderosa, flexível e perfeita para quem quer criar soluções inovadoras com responsabilidade ambiental.

Neste ebook, vamos explorar como Python pode ser usado em setores estratégicos e promover práticas sustentáveis por meio de códigos inteligentes e tecnologias verdes.



01

*Principais Setores Onde
Python Brilha*





1.1 Automação: Fazendo Mais com Menos

Automação é uma das maiores forças do Python, ajudando a reduzir tempo, esforço e desperdício.

Exemplo: Organizando arquivos automaticamente.

```
● ● ●                                     Python_Rasta

Importar OS

Def organizar_arquivos def (massas):
    durante arquivo em os.listdir(pasta):
        extensão '')[ -1 = arquivo extensão .fender('.')
        nova_pasta juntar(= SO nova_pasta .caminho.pasta, extensão)
        SO exist_ok=Verdadeiro.makedirs(nova_pasta, )
        ENTÃO os.caminho.join(nova_pasta, arquivo.renomear(SO.caminho.juntar(massa, arquivo),))

organizar_arquivos("Documentos")
print("Arquivos organizados por tipo!")
```

Aplicação: Ideal para empresas que precisam organizar grandes volumes de arquivos, reduzindo o desperdício de tempo.



1.2 Ciência de Dados: Entendendo o Mundo ao Redor

Python é líder na análise de dados, ajudando a tomar decisões mais conscientes.

Exemplo: Analisando o consumo de energia de máquinas.

```
● ● ● Python_Rasta
importar pandas como pd
dados = def {
    "Máquina": ["M1", "M2", "M3"],
    , 150,"Horas Trabalhadas extensão ": [120 110],
    [300"Energia Consumida (kWh) nova_pasta ": , 400, 250]
}
df = pd.DataFrame(dados)
Df["Eficiência (kWh/h)"] = df["Energia Consumida (kWh)"] / df["Horas Trabalhadas"]
print(Df)
```

Aplicação: Empresas podem usar essa análise para identificar desperdícios e otimizar processos industriais.

02

Códigos Conscientes: Python a Serviço do Meio Ambiente





2.1 Uso Eficiente de Recursos Computacionais

Reducir o consumo de memória e processamento faz parte de uma boa prática de sustentabilidade.

Exemplo: Criando listas de forma otimizada.

```
••• Python_Rasta

# Ineficiente
números = []
durante eu estou em def gama(1000):
    números.acrescentar(eu * 2)

extensão
# Eficiente nova_pasta
números = [i * 2 para i no intervalo (1000)]
print(números[:10]) # Exibindo os 10 primeiros
```

Aplicação: Usar menos memória é essencial ao processar grandes volumes de dados.



2.2 Processamento Paralelo para Ganhar Tempo

Executar tarefas simultaneamente reduz o tempo de execução e melhora a eficiência energética.

Exemplo: Processando grandes arquivos em paralelo.

```
... Python_Rasta

do pool de importação de multiprocessamento

Def processar_arquivo def (nome_arquivo):
    com f:abrir(nome_arquivo, 'r'
        return len(f.readlines()) # Conta linhas de um arquivo extensão

    nova_pasta
arquivos = ["dados1.txt", "dados2.txt" , "dados3.txt"]

    com Pool() como pool:
        resultados = pool.map(processar_arquivo, arquivos)

    print(f"Linhas processadas por arquivo: {resultados}")
```

Aplicação: Usar menos memória é essencial ao processar grandes volumes de dados.

03

Tecnologias Sustentáveis Apoiadas por Python





3.1 Automação Verde com IoT

A Internet das Coisas (IoT) permite criar soluções para economizar energia e otimizar recursos.

Exemplo: Monitorando o uso de energia em tempo real.

```
••• Python_Rasta

importar aleatório
Tempo de importação

def
Def monitorar_energia():
    durante _em gama(5): extensão
        consumo = random.uniform(10, 50) # Simula consumo em kWh nova_pasta
            print(f"Consumo atual: {consumo:.2f} kWh")
tempo.dormir(1)

monitorar_energia()
```

Aplicação: Dispositivos inteligentes podem ajustar consumo com base na demanda.



3.2 Previsão de Demandas para Evitar Desperdícios

Modelos preditivos ajudam a planejar melhor e evitar desperdícios de recursos.

Exemplo: Prevendo o consumo de energia com Machine Learning.

```
Python_Rasta

de sklearn.linear_model importar LinearRegression
importar numpy como np

def
# Dados simulados: horas vs consumo
horas = [2, 3, 4, 5].remodelar(-1, 1= Np.array([1, extensão ])
consumo = Np.array([30, 60, 90 nova_pasta , 120, 150])

modelo = Regressão linear().fit(horas, consumo)
previsão = modelo.predict([[6]]) # Consumo na 6ª hora
imprimir(f"Consumo previsto: {previsão[0]:.2f} kWh")
```

Aplicação: Planejar demandas energéticas com precisão para evitar subutilização ou excesso.

04

Green Software Development: Codando para o Futuro





4.1 Otimização com Geradores

Evite carregar grandes volumes de dados na memória, use geradores para criar dados sob demanda.

Exemplo: Gerando números de forma eficiente.

```
••• Python_Rasta
Def numeros_verdes(limite):
    durante eu estou em range(limite):
        rendimento i * 2 def
    durante numero em numeros_verdes(10): extensão
        imprimir(número, fim=" ") nova_pasta
```

Aplicação: Útil para sistemas que lidam com grandes bases de dados, como logs ou sensores.



4.2 Armazenamento Sustentável na Nuvem

Serviços de nuvem que usam energia renovável são aliados no desenvolvimento sustentável.

Exemplo: Upload de arquivos para a nuvem com AWS S3.

```
● ● ● Python_Rasta
importar boto3
s3 = boto3.client('s3')
arquivo = "relatorio_energia.txt"

extensao
# Upload para bucket S3 nova_pasta
s3.upload_file(arquivo, "bucket-sustentavel", arquivo)
print(f"{arquivo} enviado para o armazenamento sustentável!")
```

Aplicação: Use plataformas com compromisso ambiental para armazenar grandes volumes de dados.

CONCLUSÃO

Com Python, você pode ir além da programação tradicional, criando soluções que economizam recursos, otimizam processos e ajudam a construir um futuro mais verde.

A responsabilidade ambiental começa com pequenas escolhas no dia a dia do programador.

**Transforme Código
em Impacto Positivo**



RASTA PYTHON

"A programação é uma ferramenta poderosa; quando usada com consciência, pode transformar código em impacto positivo e inovação em sustentabilidade."

- *Lennon Müller*

A Jornada do **<Codigo_Consciente>**

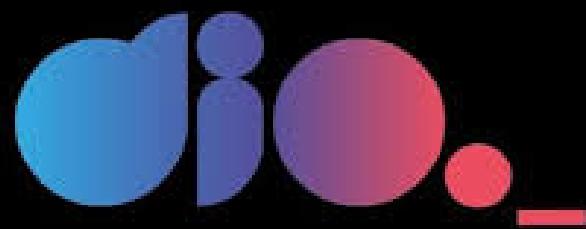
Ebook by

Lennon Müller



github.com/lennonnmueller

APOIO:



web.dio.me