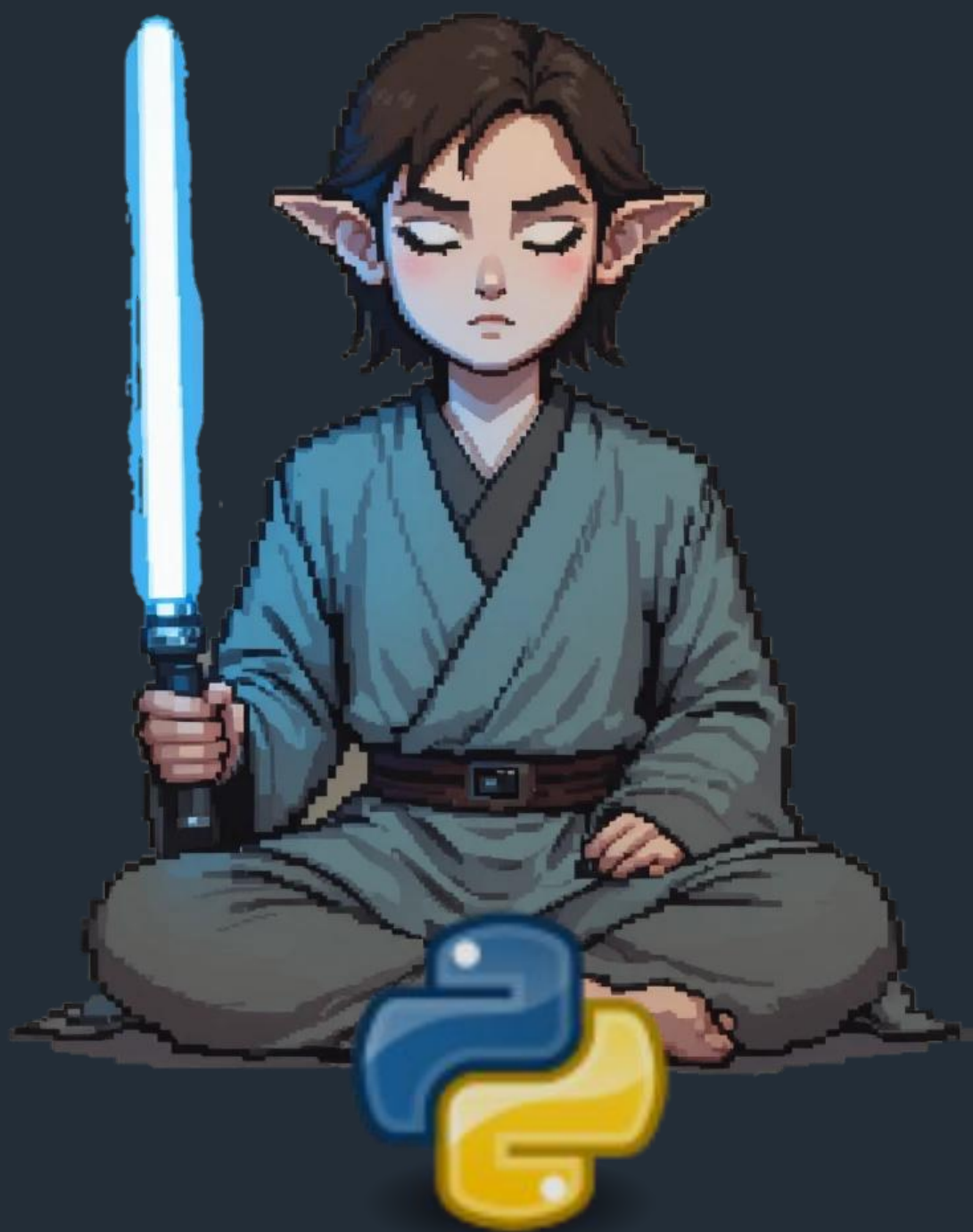


PYTHON JEDI

DOMINE A FORÇA DO PYTHON



Aprenda quais são os passos fundamentais para iniciar na linguagem Python

LUCAS SANTOS

Primeiros Passos para Dominar Python

Descubra o poder da linguagem mais versátil do mundo

Python é a porta de entrada para o universo da programação moderna.

Você pode usar Python para automação, análise de dados, IA, web, jogos e muito mais.



01

Entendendo o que é o Python



1. Entendendo o que é o Python

Simples, poderosa e feita para humanos

Python é uma linguagem criada para ser fácil de ler e entender, mesmo para iniciantes.

Sua sintaxe (modo de escrever o código) se parece com o inglês, o que a torna ideal para quem está começando.

```
print("Olá, mundo! 🚀")
```

Esse comando imprime uma mensagem na tela — é o primeiro passo para conversar com o computador.



02

Variáveis e Tipos de Dados



2. Variáveis e Tipos de Dados

Guardando informações na memória do computador

Em Python, variáveis servem para armazenar informações — como nomes, números ou valores financeiros.

O mais legal é que você não precisa declarar o tipo da variável: o Python faz isso automaticamente, tornando o processo rápido e intuitivo.

```
nome = "Ana"  
idade = 25  
saldo = 1050.75  
  
print(f"{nome} tem {idade} anos e R${saldo} na conta.")
```

Isso é um exemplo real de como exibir dados de um cliente.



2. Variáveis e Tipos de Dados

Principais tipos de dados

str → texto

int → números inteiros

float → números decimais

bool → verdadeiro ou falso

```
ativo = True  
print(type(ativo)) # Resultado: <class 'bool'>
```

Python é dinamicamente tipado, o que significa que você pode mudar o tipo da variável quando quiser.



03

Estruturas de Repetição (Loops)



3. Automatizando tarefas com loops

Automatize tarefas repetitivas com poucas linhas

Na programação, repetir tarefas manualmente não faz sentido. É aí que entram os loops, estruturas que permitem repetir ações automaticamente.

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

Isso é útil para processar listas, gerar relatórios ou enviar e-mails em lote.



3. Automatizando tarefas com loops

Exemplo contexto real

Os loops permitem repetir ações sem precisar escrever o mesmo código várias vezes.

```
contatos = ["João", "Maria", "Paula"]  
  
for pessoa in contatos:  
    print(f"Enviando e-mail para {pessoa}...")
```



04

Condições: Tomando Decisões



4. Condicionais (If, Elif, Else)

Tomando decisões inteligentes no código

Condições permitem que seu código reaja a diferentes situações, como um cérebro lógico.

```
nota = 8.5

if nota ≥ 7:
    print("Aprovado! 🎓")
else:
    print("Reprovado 😞")
```

Isso é essencial em sistemas de login, avaliações, ou regras de negócio.



4. Condicionais (If, Elif, Else)

Tomando decisões inteligentes no código

Condições permitem que seu código reaja a diferentes situações, como um cérebro lógico.

```
idade = 20
tem_carteira = True

if idade ≥ 18 and tem_carteira:
    print("Pode dirigir 🚗")
else:
    print("Não autorizado ✖")
```

Combinação de condições



05

Funções: Reutilizando Código



5. Crie seus próprios superpoderes no código

Entenda como funções tornam seu programa mais limpo e poderoso

Funções são blocos de código que realizam uma tarefa específica.

Elas tornam o programa mais organizado e reutilizável.

```
def saudacao(nome):  
    return f"Olá, {nome}! 🖐️"  
  
print(saudacao("Marcos"))
```

💡 Cada vez que você chama a função `saudacao()`, ela executa o mesmo comportamento.



06

Trabalhando com Arquivos



6. Conversando com o mundo real

Aprenda a ler e gravar arquivos com Python em poucos passos

Muitas vezes, precisamos trabalhar com arquivos — salvar relatórios, ler planilhas, registrar logs, etc.

Python torna isso extremamente fácil com a função `open()`.

```
with open("nomes.txt", "w") as arquivo:  
    arquivo.write("Alice\nBruno\nCarla")  
  
print("Arquivo criado com sucesso! ✅")
```

O `with open()` abre o arquivo e o fecha automaticamente quando o bloco termina, garantindo segurança e praticidade.

💡 Troque `"w"` por `"r"` para ler arquivos, ou por `"a"` para adicionar novas informações sem apagar o que já existe.



07

Explorando Bibliotecas



7. Expandindo seus horizontes com bibliotecas

Descubra ferramentas que fazem Python brilhar em qualquer projeto

O verdadeiro poder do Python está nas bibliotecas — pacotes prontos que ampliam suas capacidades.

```
import pandas as pd

dados = {"Nome": ["Ana", "Carlos"], "Idade": [25, 32]}
df = pd.DataFrame(dados)

print(df)
```

Quer trabalhar com dados? Use **pandas**.

Quer fazer gráficos? Experimente **matplotlib**.

Quer automatizar o navegador? **selenium** é seu amigo!

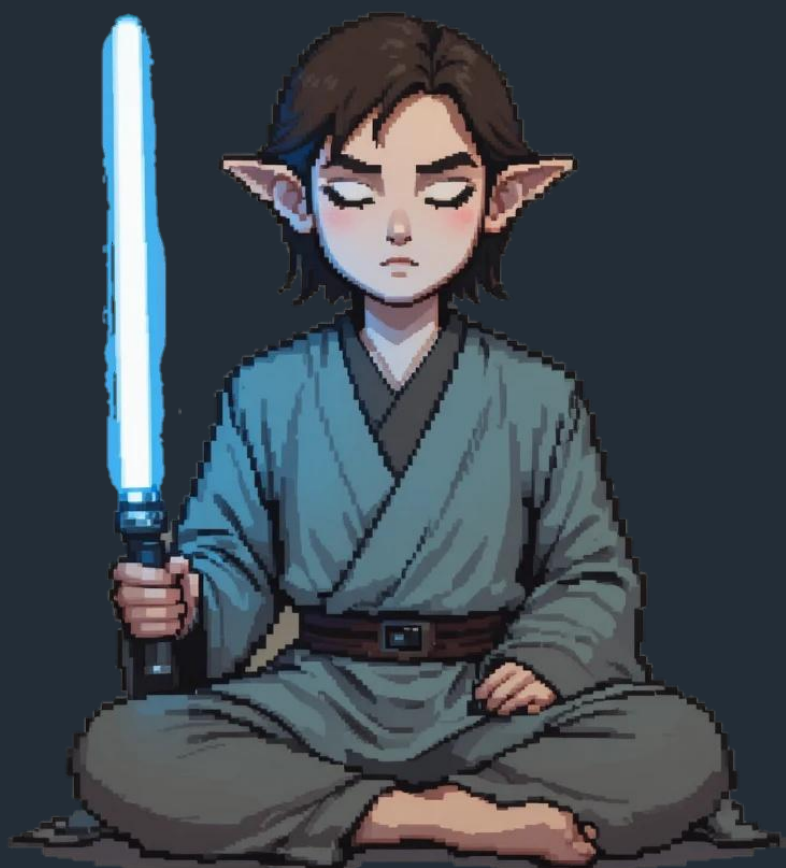
Chegando até aqui, você já domina os conceitos essenciais de Python.

O segredo agora é praticar todos os dias: refaça os exemplos, crie seus próprios scripts e explore novas ideias.



PARABÉNS

OBRIGADO POR LER ATÉ AQUI!



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

Não foi realizada uma validação cuidadosa humana, portanto, pode conter erros gerados por IA.

O ebook foi desenvolvido como parte do Bootcamp Universia Fundamentos de IA Generativa em parceria com a DIO e Santander.



AGRADECIMENTOS

