MESTRE DO PYTHON

A JORNADA DO APRENDIZ AO MESTRE DO CÓDIGO

ADQUIRA OS CONHECIMENTOS INICIAIS PARA A JORNADA PYTHON E SE TORNO UM MESTRE DO CÓDIGO

LUIZ RUGGERI

INTRODUÇÃO

A JORNADA COMEÇA

Nesta seção, apresentaremos o Python e sua importância no mundo do desenvolvimento back-end. Explicaremos de maneira simples e direta, com exemplos de código em contextos reais.

INTRODUÇÃO

Bem-vindo à sua jornada no mundo mágico do Python! Este e-book é o seu mapa para a terra encantada do desenvolvimento back-end, onde o Python é a linguagem mágica que nos permite criar, inovar e resolver problemas complexos. Vamos explorar a sintaxe básica, os tipos de dados e os operadores. Cada conceito será explicado de maneira simples e direta, com exemplos de código em contextos reais para ajudá-lo a entender melhor. Ao longo desta jornada, você não estará sozinho. Caminhando com você a cada passo do percurso, guiando-o através dos desafios e ajudando-o a superar os obstáculos. Então, pegue sua espada de código e seu escudo de lógica, e vamos embarcar nesta aventura épica juntos!

CAPÍTULO 1: O BÁSICO DE PYTHON

1.1 SINTAXE BÁSICA

Python é uma linguagem de programação de alto nível com uma sintaxe clara e concisa. Vamos começar com um exemplo simples de código que imprime “Olá, Mundo!” na tela:

Ola_mundo.py

```
print("Olá, Mundo!")
```

CAPÍTULO 1: O BÁSICO DE PYTHON

1.2 VARIÁVEIS E TIPOS DE DADOS

Em Python, as variáveis não precisam ser declaradas antes de serem usadas e os tipos de dados são inferidos automaticamente. Aqui está um exemplo de como definir uma variável e imprimir seu valor e tipo:

Declarar_variavel.py

```
idade = 25
print(idade)
print(type(idade))
```

CAPÍTULO 1: O BÁSICO DE PYTHON

1.3 OPERADORES

Python suporta uma variedade de operadores, incluindo operadores aritméticos, de comparação e lógicos. Aqui está um exemplo de como eles podem ser usados::

Operadores.py

```python
# Operadores aritméticos
soma = 5 + 3
print(soma)

# Operadores de comparação
comparacao = 5 > 3
print(comparacao)

# Operadores lógicos
logico = True and False
print(logico)
```

CAPÍTULO 2: FUNÇÕES E MÓDULOS EM PYTHON

2.1 FUNÇÕES EM PYTHON

As funções são blocos de código reutilizáveis que realizam uma tarefa específica. Aqui está um exemplo de uma função que calcula o fatorial de um número, uma operação comum em muitos problemas de programação e ciência de dados:

Funcao.py

```python
def fatorial(n):
    if n == 0:
        return 1
    else:
        return n * fatorial(n-1)

print(fatorial(5))  # Saída: 120
```

CAPÍTULO 2: FUNÇÕES E MÓDULOS EM PYTHON

2.2 MÓDULOS EM PYTHON

As funções são blocos de código reutilizáveis que realizam uma tarefa específica. Aqui está um exemplo de uma função que calcula o fatorial de um número, uma operação comum em muitos problemas de programação e ciência de dados:

Modulo.py

```python
import math

# Calculando a raiz quadrada
print(math.sqrt(16))  # Saída: 4.0

# Calculando o logaritmo
print(math.log(100, 10))  # Saída: 2.0
```

CAPÍTULO 3: TRABALHANDO COM DADOS EM PYTHON

3.1 LISTAS EM PYTHON

As listas são uma das estruturas de dados mais usadas em Python. Elas podem conter itens de diferentes tipos e são mutáveis, o que significa que você pode adicionar, remover e alterar itens após a lista ser criada. Aqui está um exemplo de como usar listas para armazenar e manipular uma série de números:

Lista.py

```python
# Criando uma lista
numeros = [1, 2, 3, 4, 5]

# Adicionando um item à lista
numeros.append(6)
print(numeros)  # Saída: [1, 2, 3, 4, 5, 6]

# Removendo um item da lista
numeros.remove(1)
print(numeros)  # Saída: [2, 3, 4, 5, 6]
```

CAPÍTULO 3: TRABALHANDO COM DADOS EM PYTHON

3.2 DICIONÁRIOS EM PYTHON

Os dicionários em Python são uma estrutura de dados que armazena pares de chave-valor. Eles são muito úteis quando você precisa associar valores a chaves para buscar rapidamente. Aqui está um exemplo de como usar dicionários para armazenar e recuperar informações sobre um livro:

Dicionario.py

```python
# Criando um dicionário
livro = {
    "titulo": "Aprendendo Python",
    "autor": "John Doe",
    "ano": 2024
}

# Acessando um valor no dicionário
print(livro["titulo"])  # Saída: Aprendendo Python

# Alterando um valor no dicionário
livro["ano"] = 2025
print(livro)  # Saída: {'titulo': 'Aprendendo Python', 'autor': 'John Doe', 'ano': 2025}
```

CAPÍTULO 3: TRABALHANDO COM DADOS EM PYTHON

3.3 ESTRUTURAS DE CONTROLE DE FLUXO

As estruturas de controle de fluxo permitem que você controle o fluxo de execução do seu programa. Aqui está um exemplo de como usar a estrutura de controle de fluxo if-else para verificar se um número é par ou ímpar:

Controle_fluxo.py

```
# Definindo um número
numero = 10

# Verificando se o número é par ou ímpar
if numero % 2 == 0:
    print("O número é par.")
else:
    print("O número é ímpar.")
```

CAPÍTULO 4: AVANÇANDO COM PYTHON

4.1 CLASSES E OBJETOS EM PYTHON

Python é uma linguagem de programação orientada a objetos. As classes fornecem uma maneira de agrupar dados e funções relacionadas. Aqui está um exemplo de como definir uma classe e criar um objeto dessa classe:

Classe.py

```python
class Carro:
    def __init__(self, marca, modelo, ano):
        self.marca = marca
        self.modelo = modelo
        self.ano = ano

    def detalhes_do_carro(self):
        return f"{self.marca} {self.modelo}, {self.ano}"

meu_carro = Carro("Toyota", "Corolla", 2020)
print(meu_carro.detalhes_do_carro())  # Saída: Toyota Corolla, 2020
```

CAPÍTULO 4: AVANÇANDO COM PYTHON

4.2 MANIPULAÇÃO DE ARQUIVOS EM PYTHON

Python fornece várias funções para ler e escrever arquivos. Aqui está um exemplo de como ler um arquivo de texto e imprimir seu conteúdo:

Manipulacao_arquivo.py

```
with open("meu_arquivo.txt", "r") as arquivo:
    conteudo = arquivo.read()
print(conteudo)
```

CAPÍTULO 4: AVANÇANDO COM PYTHON

4.3 TRATAMENTO DE EXCEÇÕES EM PYTHON

O tratamento de exceções em Python é feito com os blocos try e except. Aqui está um exemplo de como tratar uma exceção ao tentar dividir por zero::

Try_except.py

```
try:
    resultado = 10 / 0
except ZeroDivisionError:
    print("Erro: Divisão por zero não é permitida.")
```

AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI

Este e-book foi criado por IA e diagramado por humano.
O passo a passo estará disponível em Github.

Este material foi desenvolvido para fins didáticos de construção, não foi realizado um revisão cuidadosa de seu conteúdo e pode ocorrer erros gerados por uma IA.

https://github.com/LGRuggeri/Projeto_ebook_DIO