

python

A JORNADA DO HERÓI DO CÓDIGO



Natalia Cardozo

Introdução ao Python: Simples e Prático,.

Python é uma linguagem de programação poderosa, fácil de aprender e amplamente utilizada. Neste e-book, vamos explorar os conceitos fundamentais de Python com exemplos práticos e diretos, ajudando você a aplicar os conceitos no mundo real.



01

O QUE É PYTHON?



UMA LINGUAGEM PARA TODOS OS PROPÓSITOS

Python é uma linguagem de programação de alto nível conhecida por sua simplicidade e eficiência. Ela pode ser usada em aplicações web, ciência de dados, automação, inteligência artificial e muito mais.

Por que escolher Python?

- Sintaxe simples e intuitiva
- Grande comunidade de suporte
- Biblioteca padrão extensa

Vamos escrever um programa simples que exibe uma mensagem:



Primeiro comando em Python

```
### Exibe uma mensagem na tela  
print("Bem-vindo ao mundo do Python!"))
```

02

VARIÁVEIS E TIPOS DE DADOS



GUARDANDO INFORMAÇÕES

Variáveis são usadas para armazenar valores que podem ser reutilizados. Em Python, você não precisa especificar o tipo da variável.

Tipos de dados comuns:

- int: Números inteiros
- float: Números decimais
- str: Texto (strings)
- bool: Valores lógicos (True/False)

Exemplo prático:

```
● ● ● Variáveis e Tipos de Dados

# Declarando variáveis
nome = "João"
idade = 25
altura = 1.75
casado = False

# Exibindo informações
print(f"Nome: {nome}, Idade: {idade}, Altura: {altura}m, Casado: {casado}")
```

03

ESTRUTURAS CONDICIONAIS



TOMANDO DECISÕES NO CÓDIGO

Usamos condicionais para executar diferentes blocos de código com base em condições.

Operadores lógicos:

- ==: Igual
- !=: Diferente
- >: Maior que
- <: Menor que
- >=: Maior ou igual
- <=: Menor ou igual

Exemplo prático:

```
Estruturas Condicionais

# Verificar maioridade
idade = 18
if idade >= 18:
    print("Você é maior de idade.")
else:
    print("Você é menor de idade.")

# Verificar condições múltiplas
nota = 85
if nota >= 90:
    print("Aprovado com excelência!")
elif nota >= 70:
    print("Aprovado.")
else:
    print("Reprovado.")
```


04

LAÇOS DE REPETIÇÃO



AUTOMATIZANDO TAREFAS REPETITIVAS

Os laços permitem executar um bloco de código várias vezes. Os dois mais comuns em Python são:

- **for**
- **while**.

Exemplo prático:

```
Laços de Repetição

# Imprimir números de 1 a 5
for i in range(1, 6):
    print(i)

# Contagem regressiva
contador = 5
while contador > 0:
    print(contador)
    contador -= 1

# Iterar sobre uma lista
frutas = ["maçã", "banana", "laranja"]
for fruta in frutas:
    print(f"Eu gosto de {fruta}.")
```

05

FUNÇÕES



ORGANIZANDO CÓDIGO EM BLOCOS REUTILIZÁVEIS

Funções permitem agrupar códigos que realizam uma tarefa específica. Você também pode adicionar parâmetros e retornar valores.

Exemplo prático:

```
Função

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

# Chamando a função
mensagem = saudacao("Maria")
print(mensagem)

# Função com vários parâmetros
def soma(a, b):
    return a + b

resultado = soma(10, 20)
print(f"O resultado da soma é {resultado}.")
```

06

TRABALHANDO COM LISTAS



ARMAZENANDO VÁRIOS VALORES

Listas permitem armazenar uma coleção de itens em uma única variável. Elas são mutáveis, o que significa que você pode modificar seu conteúdo.

Exemplo prático:

```

Listas

# Criando uma lista
frutas = ["maçã", "banana", "laranja"]

# Adicionando e removendo itens
frutas.append("uva")
frutas.remove("banana")

# Percorrendo a lista
for fruta in frutas:
    print(fruta)

# Acessando itens diretamente
print(f"A primeira fruta é {frutas[0]}")
```

07

MANIPULANDO ARQUIVOS



LENDO E ESCRREVENDO DADOS

Python facilita a manipulação de arquivos de texto, permitindo armazenar e recuperar informações de forma eficiente

Exemplo prático:

```
Arquivo

## Escrevendo em um arquivo
with open("exemplo.txt", "w") as arquivo:
    arquivo.write("Este é um exemplo de escrita em arquivo.\n")
    arquivo.write("Python facilita o trabalho com arquivos.")

# Lendo o arquivo
with open("exemplo.txt", "r") as arquivo:
    conteudo = arquivo.read()
    print("Conteúdo do arquivo:")
    print(conteudo)
```

Conclusão

Este e-book fornece possui intuito de fornecer uma base para começar a programar em Python. Pratique cada exemplo e experimente modificar os códigos para entender melhor como tudo funciona. Ao explorar mais, você descobrirá o quão poderosa essa linguagem pode ser. Bons estudos e Boa sorte!

