

PENSANDO COMO UM PROGRAMADOR

LÓGICA DE PROGRAMAÇÃO PARA INICIANTE

By Chico Alive

A woman's profile is shown in silhouette against a white background. Overlaid on her face and hair is a blue, semi-transparent grid of binary code (0s and 1s).

Sumário


3. Introdução à Lógica de Programação
4. Variáveis e Tipos de Dados
5. Operadores e Expressões
6. Estruturas Condicionais
7. Estruturas de Repetição
8. Funções e Modularização
9. Listas e Coleções
10. Automatizando Tarefas do Dia a Dia
11. Aplicações Práticas: Cálculos Financeiros
12. Aplicações Práticas: Validação de Dados
13. Boas Práticas e Próximos Passos
- 14. Sobre o Autor**



Capítulo 1: Introdução à Lógica de Programação

- **O que é lógica de programação?**
- Lógica de programação é a base para resolver problemas com o uso do computador. Ela envolve pensar de forma estruturada para escrever instruções claras que o computador entenda.
- **Por que aprender lógica?**
- Aprender lógica de programação ajuda você a:
- Resolver problemas com mais clareza.
- Automatizar tarefas repetitivas.
- Criar programas simples e úteis para o dia a dia.
- **Exemplo prático:**
- Imagine que você quer lembrar de regar as plantas a cada dois dias. Com lógica de programação, você pode automatizar esse lembrete.

python


 Copiar código

```
dia_atual = 6 # Suponha que hoje seja dia 6
if dia_atual % 2 == 0:
    print("Hoje é dia de regar as plantas!")
else:
    print("Hoje não precisa regar.")
```

Capítulo 2: Variáveis e Tipos de Dados

- **O que são variáveis?**
- Variáveis são espaços na memória para guardar informações.
- **Tipos comuns de dados:**
- **Inteiro (int):** números inteiros (ex: 10, -3)
- **Decimal (float):** números com vírgula (ex: 3.14)
- **Texto (str):** cadeias de caracteres (ex: "Olá")
- **Booleano (bool):** verdadeiro ou falso
- **Exemplo prático: calcular idade**

python

 Copiar código

```
ano_nascimento = 2000
ano_atual = 2025
idade = ano_atual - ano_nascimento
print("Você tem", idade, "anos.")
```

Capítulo 3: Operadores e Expressões

Operadores matemáticos:


- Soma: +
- Subtração: -
- Multiplicação: *
- Divisão: /
- Módulo (resto): %

Operadores lógicos:

- Igual: ==
- Diferente: !=
- Maior: >
- Menor: <
- E: and
- Ou: or

Exemplo prático: verificar saldo

python

 Copiar código

```
saldo = 150
compra = 100
if saldo >= compra:
    print("Compra aprovada.")
else:
    print("Saldo insuficiente.")
```


Capítulo 4: Estruturas Condicionais

O que é uma condição?

É uma decisão baseada em uma pergunta do tipo "sim ou não".

Estrutura básica:


python

 Copiar código

```
if condição:
    # faz algo
else:
    # faz outra coisa
```

Exemplo prático: verificar se o número é par ou ímpar

python

 Copiar código


```
numero = 7
if numero % 2 == 0:
    print("Número par.")
else:
    print("Número ímpar.")
```

Capítulo 5: Estruturas de Repetição

- Para que servem?
- Para repetir um bloco de código várias vezes.

while: repete enquanto a condição for verdadeira


python

 Copiar código

```
contador = 1
while contador <= 5:
    print("Contando:", contador)
    contador += 1
```

for: repete para cada item em uma sequência

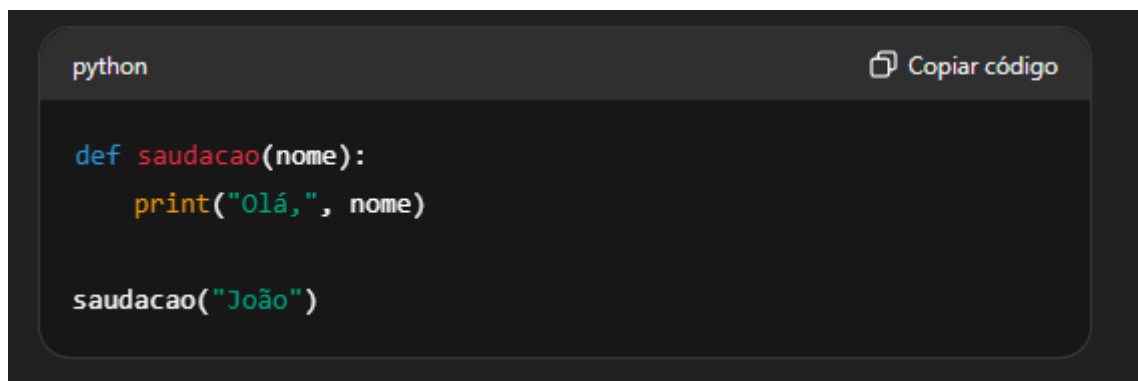
python

 Copiar código

```
for i in range(1, 6):
    print("Contando:", i)
```

Capítulo 6: Funções e Modularização

- **O que são funções?**
- Funções agrupam comandos e facilitam a reutilização



A screenshot of a code editor with a dark theme. The editor shows a Python function definition and its call. The function is named `saudacao` and takes a parameter `nome`. It prints "Olá," followed by the value of `nome`. The function is then called with the argument "João". In the top right corner of the editor, there is a button labeled "Copiar código" with a copy icon.

```
python

def saudacao(nome):
    print("Olá,", nome)


saudacao("João")
```

- **Vantagens:**
- Reutilização de código
- Organização
- Legibilidade

Capítulo 7: Listas e Coleções

- O que são listas?
- Conjunto de valores armazenados em sequência


python

 Copiar código

```
frutas = ["maçã", "banana", "laranja"]  
print(frutas[0]) # mostra "maçã"
```

Iterando sobre listas:

python


 Copiar código

```
for fruta in frutas:  
    print("Gosto de", fruta)
```

Capítulo 8: Automatizando Tarefas do Dia a Dia

- Exemplo: enviar lembretes de vencimento

python


 Copiar código

```
contas = ["Luz", "Água", "Internet"]  
for conta in contas:  
    print("Lembrete: pagar", conta)
```

Capítulo 9: Aplicações Práticas: Cálculos Financeiros

Exemplo: simular rendimento de investimento

python

 Copiar código

```
capital = 1000
taxa = 0.01 # 1% ao mês
meses = 12


for mes in range(1, meses + 1):
    capital += capital * taxa

print("Valor após 12 meses:", round(capital, 2))
```

Capítulo 10: Aplicações Práticas: Validação de Dados

Verificar se CPF tem 11 dígitos

python



 Copiar código

```
cpf = "12345678901"
if len(cpf) == 11 and cpf.isdigit():
    print("CPF válido.")
else:
    print("CPF inválido.")
```

Capítulo 11: Boas Práticas e Próximos Passos

- **Boas práticas:**
- Nomear variáveis de forma clara
- Comentar o código quando necessário
- Testar cada parte do programa
- **E agora?**
- Você pode:
- Aprender uma linguagem de programação como Python, JavaScript ou C#
- Criar pequenos projetos (calculadoras, bots, automações)
- Participar de comunidades e fóruns

Sobre o Autor

- Olá! Meu nome é **Francisco**. Sou formado em **Design Gráfico** e em **Análise e Desenvolvimento de Sistemas** pela **Estácio**. Atualmente, trabalho como **tutor na Cruzeiro do Sul**.
- Estou em constante aprendizado e recentemente mergulhei no universo da programação com o objetivo de **impulsionar minha carreira** e expandir minhas habilidades técnicas.
- Se quiser acompanhar minha trajetória ou trocar ideias sobre desenvolvimento, design ou educação, fique à vontade para me seguir nas redes:
-  [LinkedIn: chicoalive](#)
-  [GitHub: chicoalive](#)
- Vamos crescer juntos!