

LÓGICA DO VIGILANTE DE GOTHAM:

"DOMINANDO A PROGRAMAÇÃO
COM A PRECISÃO DO BATMAN"



EVERTON RODRIGUES

LÓGICA DO VIGILANTE DE GOTHAM:

Dominando a Programação com a Lógica do Batman

Bem-vindo ao “**Logica do Vigilante de Gotham**”.

Neste eBook, você aprenderá a dominar a programação utilizando a lógica de forma simples e eficiente. Inspirado pelo Batman, um dos maiores estrategistas, vamos explorar os fundamentos da lógica na programação com exemplos práticos e soluções de problemas reais. Vamos começar!



01

ENTENDENDO A LÓGICA NA PROGRAMAÇÃO

A lógica na programação é a base de qualquer aplicativo ou sistema. Compreender como resolver problemas de forma lógica é essencial para desenvolver códigos eficientes e funcionais.

ENTENDENDO A LÓGICA NA PROGRAMAÇÃO

A lógica na programação é a base de qualquer aplicativo ou sistema. Compreender como resolver problemas de forma lógica é essencial para desenvolver códigos eficientes e funcionais.



02

ESTRUTURAS CONDICIONAIS: O "SE" E O "SENÃO" DA LÓGICA

As estruturas condicionais permitem que você execute diferentes blocos de código com base em certas condições.

ESTRUTURAS CONDICIONAIS: O "SE" E O "SENÃO" DA LÓGICA

As estruturas condicionais permitem que você execute diferentes blocos de código com base em certas condições.

Exemplo:

```
Logica

idade = 20
if idade >= 18:
    print("Você é maior de idade.")
else:
    print("Você é menor de idade.")
```

Resolução: Aqui, verificamos se a idade é maior ou igual a 18. Se for, imprimimos uma mensagem; caso contrário, imprimimos outra.



03

LOOPS: REPETIÇÃO INTELIGENTE

Loops são usados para repetir um bloco de código várias vezes.

LOOPS: REPETIÇÃO INTELIGENTE

Loops são usados para repetir um bloco de código várias vezes.

Exemplo:

```
Logica
for i in range(5):
    print(f"Treinamento do Batman dia {i+1}")
```

Resolução: Neste exemplo, o loop for repete o print cinco vezes, incrementando o valor de i a cada iteração.



04

FUNÇÕES: MODULARIZANDO A LÓGICA

Funções permitem que você encapsule blocos de código reutilizáveis.

FUNÇÕES: MODULARIZANDO A LÓGICA

Funções permitem que você encapsule blocos de código reutilizáveis.

Exemplo:

```
Logica
def verificar_maioridade(idade):
    if idade >= 18:
        return "Maior de idade"
    else:
        return "Menor de idade"
resultado = verificar_maioridade(20)
print(resultado)
```

Resolução: Criamos uma função verificar_maioridade que retorna se a pessoa é maior ou menor de idade. Chamamos a função com um valor de idade e imprimimos o resultado.



05

ESTRUTURAS DE DADOS: ORGANIZANDO INFORMAÇÕES

Estruturas de dados como listas, tuplas e dicionários ajudam a organizar e manipular informações de forma eficiente.

ESTRUTURAS DE DADOS: ORGANIZANDO INFORMAÇÕES

Estruturas de dados como listas, tuplas e dicionários ajudam a organizar e manipular informações de forma eficiente.

Exemplo:

```
Logica
gadgets_batman = ["Batarang", "Batmobile", "Bat-signal"]
for gadget in gadgets_batman:
    print(f"Batman usa: {gadget}")
)
```

Resolução: Usamos uma lista para armazenar gadgets e um loop for para iterar e imprimir cada um.



06

DEPURAÇÃO: ENCONTRANDO E CORRIGINDO ERROS

Depurar é o processo de encontrar e corrigir erros no código.

DEPURAÇÃO: ENCONTRANDO E CORRIGINDO ERROS

Depurar é o processo de encontrar e corrigir erros no código.

Exemplo:

```
Logica
def dividir(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return "Erro: Divisão por zero não é permitida."
resultado = dividir(10, 0)
print(resultado)
```

Resolução: Usamos um bloco try-except para capturar e tratar um possível erro de divisão por zero.



07

APLICAÇÕES PRÁTICAS: RESOLVENDO PROBLEMAS REAIS

Para finalizar, vamos aplicar a lógica aprendida em um contexto real.

APLICAÇÕES PRÁTICAS: RESOLVENDO PROBLEMAS REAIS

Para finalizar, vamos aplicar a lógica aprendida em um contexto real.

Exemplo: Calculadora Simples

```
Untitled-1

def calculadora(a, b, operacao):
    if operacao == "+":
        return a + b
    elif operacao == "-":
        return a - b
    elif operacao == "*":
        return a * b
    elif operacao == "/":
        return a / b
    else:
        return "Operação inválida."

resultado = calculadora(10, 5, "+")
print(f"Resultado: {resultado}")
```

Resolução: Criamos uma função de calculadora que executa operações básicas com base na entrada fornecida.

AGRADECIMENTOS:



OBRIGADO!

Esse Ebook foi gerado por IA, diagramado e
detalhado por humano.
Mais informações segue no Github abaixo.

“Esse conteúdo foi criado com ajuda e apoio didático
da equipe da DIO e Felipe Aguiar.”



<https://github.com/EvertonOx/Ebook-Logica-do-Batman>

