

Curso lógica de programação

CODE - X



Lógica de programação - Python



PALAVRA DO ORIENTADOR

Olá! Sou formado nas áreas técnicas de Química, Meio Ambiente e Tecnologia da Informação, além de possuir graduação em Licenciatura em Matemática. Minha paixão pelo conhecimento e pelo ensino sempre me motivou a buscar a excelência nas minhas áreas de atuação. Ao longo da minha jornada, tive a vantagem de acumular experiências valiosas, tanto no campo profissional quanto acadêmico, e acredito que o aprendizado é um processo contínuo e enriquecedor.

É com grande satisfação que ofereço apoio e orientação a você, que está começando ou se aprofundando em sua carreira e nos estudos. Compreendo os desafios e as conquistas desta fase e fico muito feliz por poder compartilhar um pouco da minha trajetória e conhecimento.

Sinta-se à vontade para contar comigo durante essa jornada intensa de estudos e trabalho. Estou aqui para ajudar no que for necessário, oferecendo orientação, apoio e incentivo. Juntos, podemos transformar desafios em oportunidades de crescimento e alcançar grandes objetivos. Conte sempre comigo!

SUMÁRIO

Introdução.....	05
1. Variáveis e tipos de dados.....	07
2. Operadores	08
3. Estruturas Condicionais	09
4. Estruturas de Repetição (Loops)	09
5. Funções	10
6. Estruturas de Dados	11
7. Manipulação de Strings	11
8. Entrada e Saída de Dados	12
9. Tratamento de Erros (Exceções)	12
10. Módulos e Bibliotecas	12
21 Referências	14

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO

Olá, estudante! Seja muito bem-vindo ao fascinante universo da programação! Se você está aqui, é porque, em algum momento, já ouviu falar sobre códigos, softwares, linguagens de programação e, claro, sobre a essencial Lógica de Programação. Mas talvez você esteja se perguntando: “Por onde começar?” Não se preocupe, estou aqui para guiá-lo nessa jornada inicial.

Nesta unidade, vamos explorar juntos os fundamentos da lógica de programação utilizando a linguagem Python. Python é uma das linguagens mais populares e versáteis do mercado, conhecida por sua simplicidade e eficiência, tornando-a uma excelente escolha para quem está começando. Você descobrirá como conceitos fundamentais como variáveis, estruturas condicionais, laços de repetição e funções são essenciais para construir qualquer tipo de software, do mais simples ao mais complexo.

Este conjunto de conhecimentos básicos não apenas o preparará para enfrentar os desafios do mercado de trabalho com confiança, mas também o capacitará a pensar de forma lógica e estruturada. A lógica de programação é a chave para resolver problemas de forma eficiente, e dominá-la permitirá que você contribua de maneira significativa para o avanço tecnológico, criando soluções inovadoras e práticas.

Prepare-se para dar os primeiros passos rumo ao seu desenvolvimento como programador!

Introdução

A lógica de programação é a base de todo desenvolvimento em tecnologia e o ponto de partida para qualquer pessoa que deseje aprender a programar. Para quem está iniciando, dominar a lógica de programação é essencial porque ela permite compreender como os computadores "pensam" e executam comandos. A lógica ensina a organizar o raciocínio, de modo a resolver problemas de forma clara e eficiente, utilizando passos simples e estruturados. Este aprendizado inicial cria uma base sólida que facilita a transição para linguagens específicas, tornando menos desafiador o estudo de novos códigos e estruturas.

Para aqueles que já têm alguma experiência e desejam aprimorar suas habilidades, uma lógica de programação é igualmente crucial. A lógica refinada não apenas melhorou a qualidade do código, mas também tornou o processo de desenvolvimento mais ágil e eficiente. Programadores experientes sabem que dominar a lógica de programação ajuda a evitar erros comuns, a criar algoritmos mais eficazes e a desenvolver soluções que consomem menos recursos. A habilidade de pensar logicamente permite ao programador enfrentar desafios mais complexos e criar soluções mais criativas e otimizadas. Seja para iniciantes ou para quem deseja aprimorar suas habilidades, a lógica de programação é o aprendizado sobre o que se constrói a capacidade de transformar ideias em realidade, através de códigos que funcionam.

Ela desenvolve o pensamento crítico, a capacidade analítica e a habilidade de resolver problemas de maneira estruturada e eficiente — competências que são indispensáveis no cenário atual, onde a tecnologia evolui rapidamente e exige soluções cada vez mais sofisticadas.

Portanto, investir no aprendizado da lógica de programação é investir em uma habilidade que transcende qualquer linguagem específica e que permanecerá relevante ao longo de toda a trajetória profissional em tecnologia. É a chave para se tornar um desenvolvedor completo, capaz de enfrentar qualquer desafio e criar soluções inovadoras no universo digital.



FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - PYTHON

1. Variáveis e tipos de dados

O que são: Variáveis são espaços na memória usados para armazenar valores que podem ser alterados ao longo da execução do programa. Em Python, não é necessário declarar o tipo de variável explicitamente, pois a linguagem é dinamicamente tipada.

Exemplo:

```
nome = "João"          # Variável do tipo string
idade = 25              # Variável do tipo inteiro
altura = 1.75            # Variável do tipo float
ativo = True             # Variável do tipo booleano
```

Tipos de Dados Comuns:

- int(inteiro): números inteiros, como 10, -5.
- float(decimal): números com ponto flutuante, como 3.14, -0.7.
- str(string): sequência de caracteres, como "Python".
- bool(booleano): valores lógicos, True ou False.
- list, tuple, set, dict: estruturas de dados que armazenam coleções de valores.

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - PYTHON

2. Operadores

O que são: Operadores são símbolos que executam operações sobre variáveis e valores.

Tipos de Operadores:

- Aritméticos: + (adição), -(subtração), *(multiplicação), /(divisão), //(divisão inteira), %(módulo), **(exponenciação).

```
soma = 5 + 3      # Resultado: 8  
divisao = 10 / 2    # Resultado: 5.0  
resto = 10 % 3      # Resultado: 1
```

- Comparação: == (igual a), !=(diferente de), >(maior que), <(menor que), >=(maior ou igual), <=(menor ou igual).

```
resultado = 5 > 3    # Resultado: True
```

Lógicas: and , or, not

```
verdadeiro = True and False  # Resultado: False
```

- Atribuição: = , +=, -=, *=, /=.

WWW.REALLYGREATSITE.COM

```
x = 10  
x += 5      # x agora é 15
```

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - PYTHON

3. Estruturas Condicionais

O que são: As estruturas condicionais permitem que o programa tome decisões baseadas em condições, alterando o fluxo do código de acordo com os valores das variáveis.

Principais Estruturas:

- if, elif, else.

```
idade = 18
if idade >= 18:
    print("Você é maior de idade.")
elif idade >= 13:
    print("Você é adolescente.")
else:
    print("Você é criança.")
```

4. Estruturas de Repetição (Loops)

- O que são: São usados para repetir um bloco de código várias vezes, até que uma condição esteja satisfeita.
- Principais Estruturas:
- for: usado quando se sabe o número de repetições ou para percorrer uma coleção.

```
for i in range(5):
    print(i) # Imprime de 0 a 4
```

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - PYTHON

- while: executa enquanto uma condição para verdadeira.

```
contador = 0
while contador < 5:
    print(contador)
    contador += 1
```

5. Funções

O que são: Funções são blocos de código que realizam uma tarefa específica. Eles ajudam a modularizar o código, facilitando a reutilização e manutenção.

Definindo e Chamando Funções:

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

saudacao("Maria") # Saída: Olá, Maria!
```

Parâmetros e Retorno:

```
def soma(a, b):
    return a + b

resultado = soma(5, 7) # Resultado: 12
```

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - PYTHON

6. Estruturas de Dados

O que é: Em Python, existem várias estruturas de dados para armazenar e manipular coleções de informações.

Principais Estruturas:

- Listas: coleções mutáveis de itens.

```
frutas = ["maçã", "banana", "laranja"]
frutas.append("uva") # Adiciona "uva" à lista
```

- Tuplas: coleções imutáveis.

```
coordenadas = (10, 20)
```

- Dicionários: coleções de pares chave-valor.

```
pessoa = {"nome": "João", "idade": 30}
print(pessoa["nome"]) # Saída: João
```

- Conjuntos: coleções de elementos únicos.

```
numeros = {1, 2, 3, 3} # {1, 2, 3}
```

7. Manipulação de Strings

O que é: Python tem uma manipulação poderosa de strings, permitindo a criação e modificação de textos com facilidade.

Exemplos:

```
texto = "Olá, Mundo!"
print(texto.lower()) # Saída: olá, mundo!
print(texto.upper()) # Saída: OLÁ, MUNDO!
print(texto.split(",")) # Saída: ['olá', 'Mundo!']
```

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - PYTHON

8. Entrada e Saída de Dados

O que é: A entrada permite ao usuário fornecer informações para o programa, enquanto a saída exibe dados.

Exemplo:

```
nome = input("Qual é o seu nome? ")
print(f"Olá, {nome}!")
```

9. Tratamento de Erros (Exceções)

O que é: O tratamento de erros permite lidar com situações inesperadas que podem ocorrer durante a execução do programa, evitando que ele pare de funcionar.

Exemplo:

```
try:
    numero = int(input("Digite um número: "))
    print(f"O dobro de {numero} é {numero * 2}")
except ValueError:
    print("Você deve digitar um número válido!")
```

10. Módulos e Bibliotecas

O que é: Python possui uma vasta coleção de módulos e bibliotecas que permitem adicionar funcionalidades extras ao seu programa.

Exemplo:

```
import math

raiz = math.sqrt(16)
print(raiz) # Saída: 4.0
```

FUNDAMENTOS DE LÓGICA DE PROGRAMAÇÃO - PYTHON

Estes fundamentos fornecem uma base sólida para começar a programar em Python e desenvolver habilidades avançadas no futuro. A prática constante desses conceitos permitirá que você construa programas cada vez mais sofisticados e eficientes.

CODE-X

REFERÊNCIAS

SOUZA, P. R. Desenvolvimento Web Moderno: Fundamentos e Práticas. Porto Alegre: Editora Inova, 2022.

SILVA, J. Python no desenvolvimento de back-end: Simplicidade e poder. Revista de Programação e Desenvolvimento de Software, v. 8, n. 3, p. 75-88, 2020