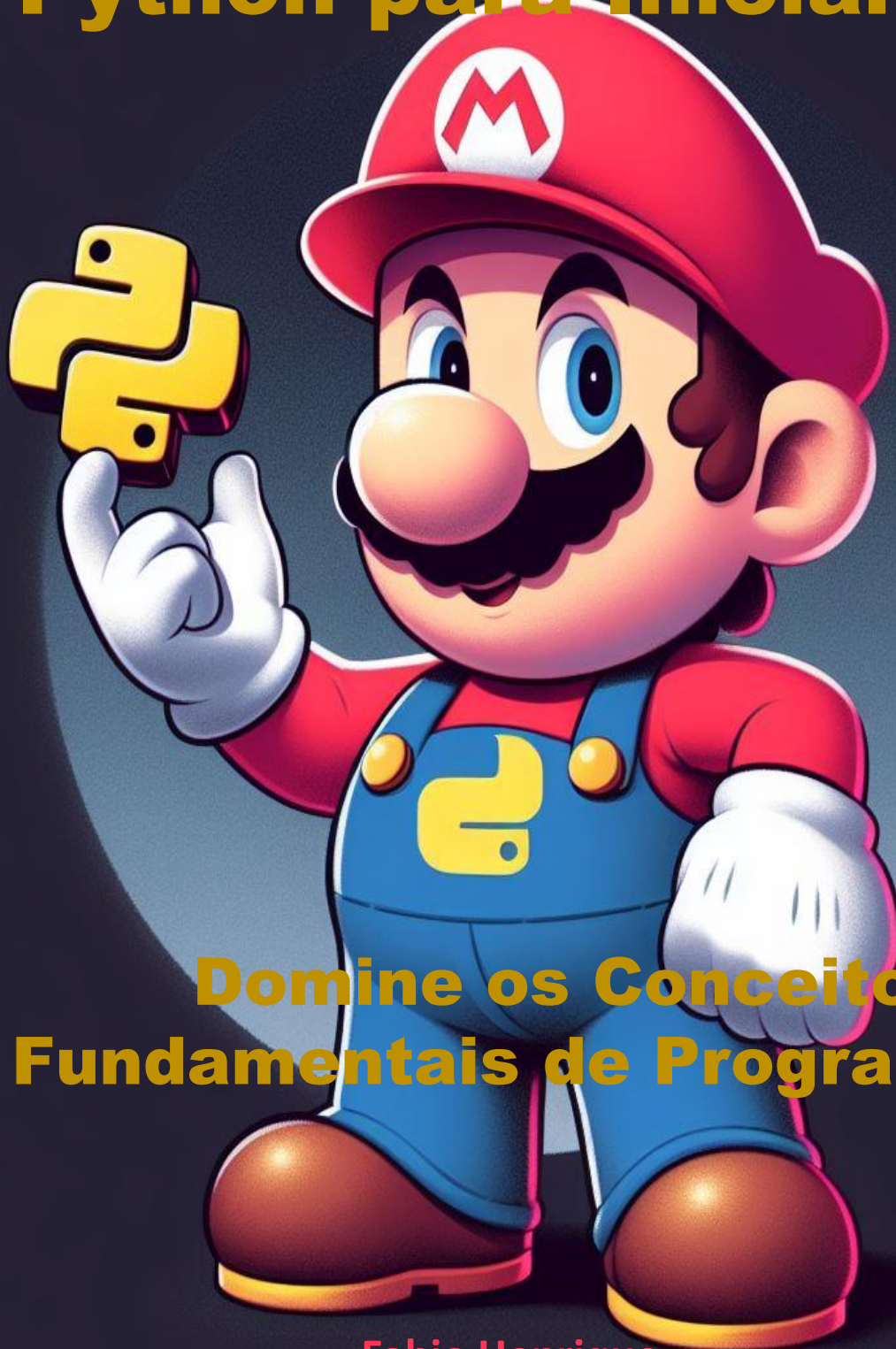


Python para Iniciantes



**Domine os Conceitos
Fundamentais de Programação**

Fabio Henrique



Introdução

Seja bem-vindo ao mundo da programação em Python! Se você é novo na codificação ou está apenas começando sua jornada de aprendizado em programação, Python é uma excelente escolha para dar os primeiros passos. Com sua sintaxe simples e legibilidade, Python é uma linguagem de programação acessível e poderosa, amplamente utilizada em uma variedade de campos, desde desenvolvimento web e científico até automação e inteligência artificial.

Neste guia introdutório, vamos explorar os conceitos fundamentais de Python, começando do zero e construindo uma base sólida para sua jornada de aprendizado.

Vamos começar!



Fundamentos

Variáveis

Para declarar variáveis em Python, basta atribuir um valor a um nome. Python é uma linguagem dinamicamente tipada, então você não precisa especificar o tipo de dados.

Em Python você pode criar uma variável atribuindo valor a ela. Por exemplo:

```
variables.py  
  
x = 10 # Inteiro  
y = 20.5 # Ponto flutuante
```



Variáveis



Você pode nomear suas variáveis de acordo com certas regras:

O nome da variável deve começar com uma letra (a-z, A-Z) ou um sublinhado (_).

O nome da variável pode conter letras, números e sublinhados.

O nome da variável não pode começar com um número.

Python faz diferenciação entre maiúsculas e minúsculas, então `nome` e `Nome` seriam duas variáveis diferentes.

Aqui estão alguns exemplos de criação de variáveis em Python:

```
python
```

```
nome = "João"  
idade = 25  
altura = 1.75  
eh_estudante = True
```


Neste exemplo, criamos quatro variáveis: `nome`, `idade`, `altura` e `eh_estudante`, com tipos diferentes de dados: Uma string, um inteiro, um número de ponto flutuante um booleano, respectivamente.



Tipos de dados

integers


São números inteiros, positivos, negativos, sem parte decimal.

 data_types.py

```
age = 25
```

floats


Representam números reais e podem incluir uma parte decimal.

 data_types.py

```
height = 175.5
```

strings

Texto entre aspas simples ou duplas.

 data_types.py

```
name = "John Doe"
```



Tipos de dados



Booleans

Valores lógicos que podem ser True ou False. São comumente usados em expressões lógicas e de comparação para avaliar condições.

```
data_types.py  
  
is_online = True  
is_online = False
```





Operadores

Operadores aritméticos

Usados para realizar operações matemáticas como adição, subtração, multiplicação e divisão.

```
operators.py

sum = 10 + 5 # 15
product = 10 * 5 # 50
```

Operadores de comparação

Usados para comparar valores.
Retorna True ou False.

```
operators.py

print(10 > 5) # True
print(10 == 5) # False
```

Operadores lógicos

Usados para combinar condições lógicas.

```
operators.py

print(10 > 5 and 10 < 20) # True
print(10 < 5 or 10 < 20) # True
```





Controle de fluxo

Declarações if-else

Permite que você execute condicionalmente blocos de código.

```
control_flow.py

if age ≥ 18:
    print("Adult")
else:
    print("Minor")
```

for Loops

Executa um bloco de códigos várias vezes, útil para iterar sobre uma sequência.

```
control_flow.py

for i in range(5):
    print(i)
```

while Loops

Repete um bloco de código enquanto uma condição é verdadeira.

```
control_flow.py

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





Funções

Definição de Funções

Você pode definir funções para encapsular códigos para reutilização.

```
functions.py

def greet(name):
    return "Hello, " + name
```

Chamando Funções

Chama uma função para executar seu código.

```
functions.py

print(greet("Alice"))
```

Parâmetros de Funções

Passa informação para funções através de parâmetros.

```
functions.py

def greet(name, message):
    return f"{message}, {name}"
```





Atribuindo valores

Retorna um valor de uma função, é uma operação fundamental em Python, usada para armazenar dados em variáveis.

```
functions.py  
  
def sum(a, b):  
    return a + b
```





Obrigado por ler até aqui.

Esse conteúdo foi gerado por IA e diagramado por um humano em aprendizado de Python.

O conteúdo foi gerado com fins didáticos e não passou por uma validação cuidadosa profissional humana. Contudo foi feito com carinho e cuidadosamente para você leitor.



fabiohenriqueagro@hotmail.com



www.linkedin.com/in/fabiohenriqueagro



Github.com/FabioAgroTechnology

