

# A Liga Python: O Despertar da Força do Desenvolvedor



Rhyan Andrade



# Introdução a linguagem

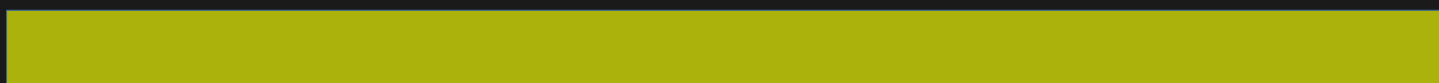
## O começo de estudos em Python

Aprender Python pode ser simples e divertido! Neste guia, você vai encontrar os fundamentos essenciais para quem está começando, sempre com explicações objetivas, exemplos reais e subtítulos para facilitar sua leitura.



# 01

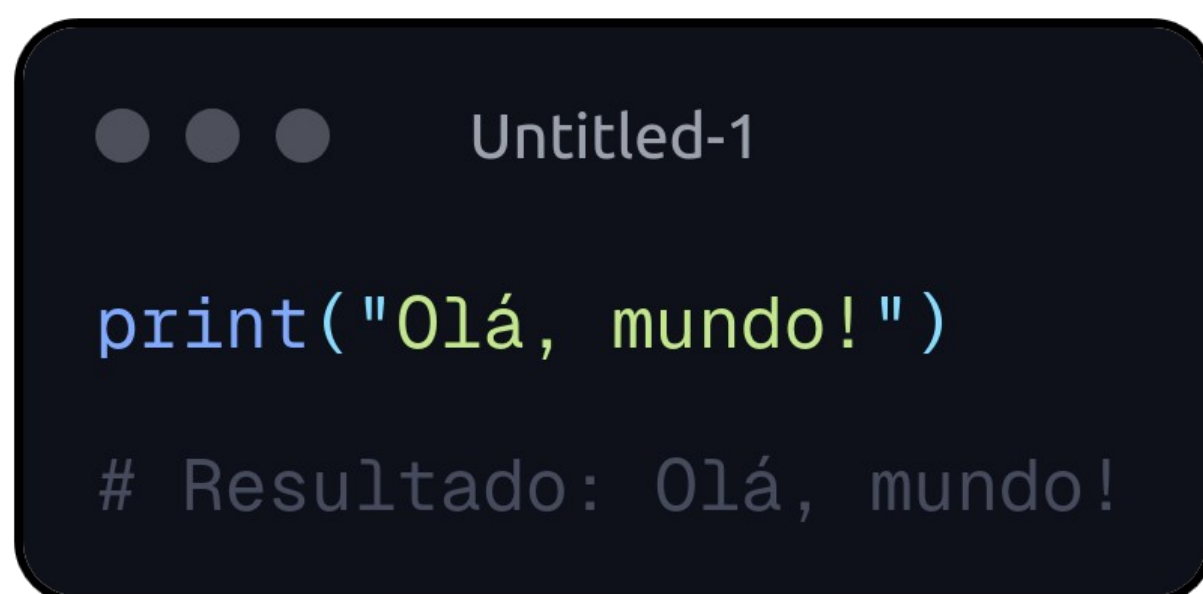
**Seu primeiro  
Hello World  
em Python**



# Imprimindo na Tela: Seu Primeiro Programa

O que é o `print()`?

A função “**print()**” serve para mostrar mensagens e resultados no console. É a primeira coisa que um programador aprende, pois permite conferir rapidamente se o código está funcionando como esperado.



```
print("Olá, mundo!")  
  
# Resultado: Olá, mundo!
```

Quando usar?

Use “**print()**” para exibir informações ao usuário, testar resultados de cálculos ou conferir valores de variáveis durante o desenvolvimento.

# 02

**Variáveis:  
Guardando  
Informações**



# Variáveis: Guardando Informações

## O que são variáveis?

Variáveis são espaços na memória onde você armazena dados temporários, como nomes, idades, resultados de contas, etc. Atribua um valor a uma variável usando o sinal “=”.



Untitled-1

```
nome = "Ana"  
idade = 25  
print("Nome:", nome, "| Idade:", idade)
```

## Quando usar?

Com variáveis, você pode reaproveitar informações e manipular dados ao longo do programa, como calcular a idade em anos futuros ou criar mensagens personalizadas.

# 03

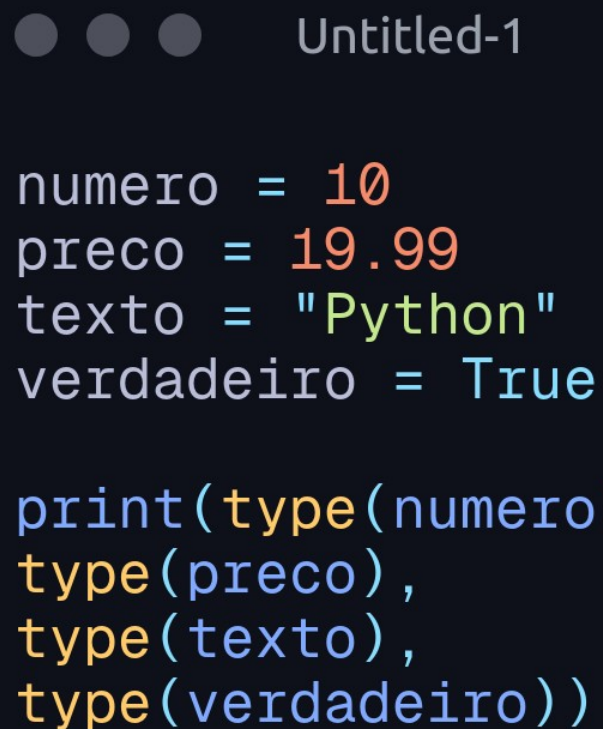
## **Tipos de Datos**

---

# Tipos de Dados

## Quais são os tipos básicos?

- int**: Números inteiros, como 10, -3 ou 0.
- float**: Números com casas decimais, como 19.99 ou -2.5.
- str**: Sequências de texto, como "Python" ou "Olá".
- bool**: Valores lógicos, True (verdadeiro) ou False (falso).



```
numero = 10
preco = 19.99
texto = "Python"
verdadeiro = True

print(type(numero),
      type(preco),
      type(texto),
      type(verdadeiro))
```

## Para que servem?

Cada tipo é usado em situações diferentes: preços (float), quantidade (int), nomes (str), verificações e condições (bool).



# 04

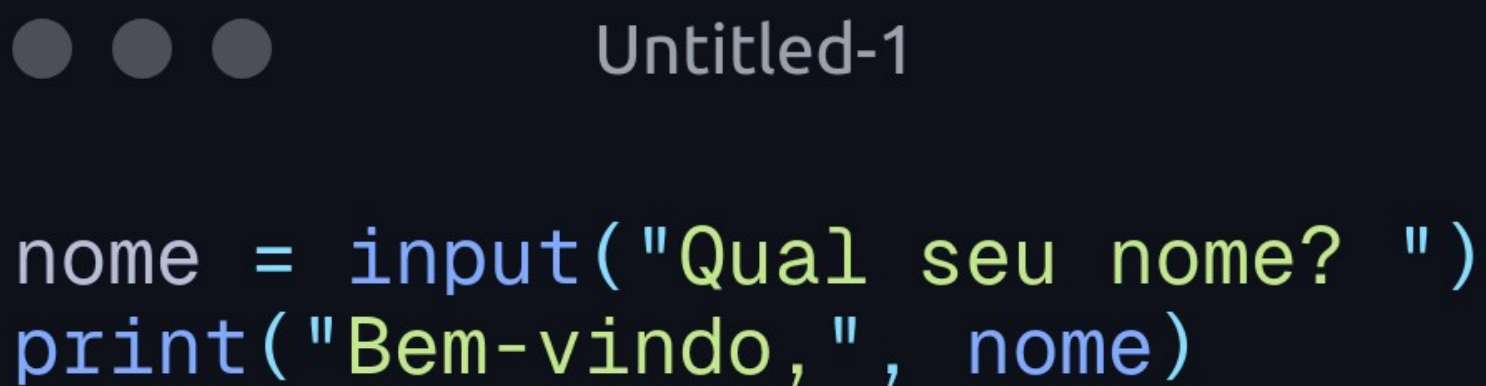
## Entrada do Usuário



# Entrada do Usuário: Conversando com o Programa

## Como receber dados?

A função **input()** permite que o usuário digite informações para o programa. Tudo que vem do **input()** é texto, então converta para outros tipos se necessário.

A dark-themed code editor window titled "Untitled-1" with three window control buttons (red, yellow, green) in the top-left corner. The code inside is written in a light blue/green monospace font.

```
nome = input("Qual seu nome? ")  
print("Bem-vindo,", nome)
```

## Quando usar?

Use quando quiser personalizar o programa, pedindo dados do usuário, como nome, idade ou preferências.

# 05

## Condicionais

---

# Condicionais: Tomando Decisões

## Como funcionam as escolhas?

Com “if”, `elif` e `else`, você faz o programa executar ações diferentes conforme condições.



Untitled-1

```
idade = int(input("Digite sua idade: "))  
if idade ≥ 18:  
    print("Você é maior de idade.")  
else:  
    print("Você é menor de idade.")
```

## Para que servem?

Permitem criar programas inteligentes, que respondem de acordo com a situação do usuário (ex: liberar acesso, aplicar descontos, etc.).



# 06

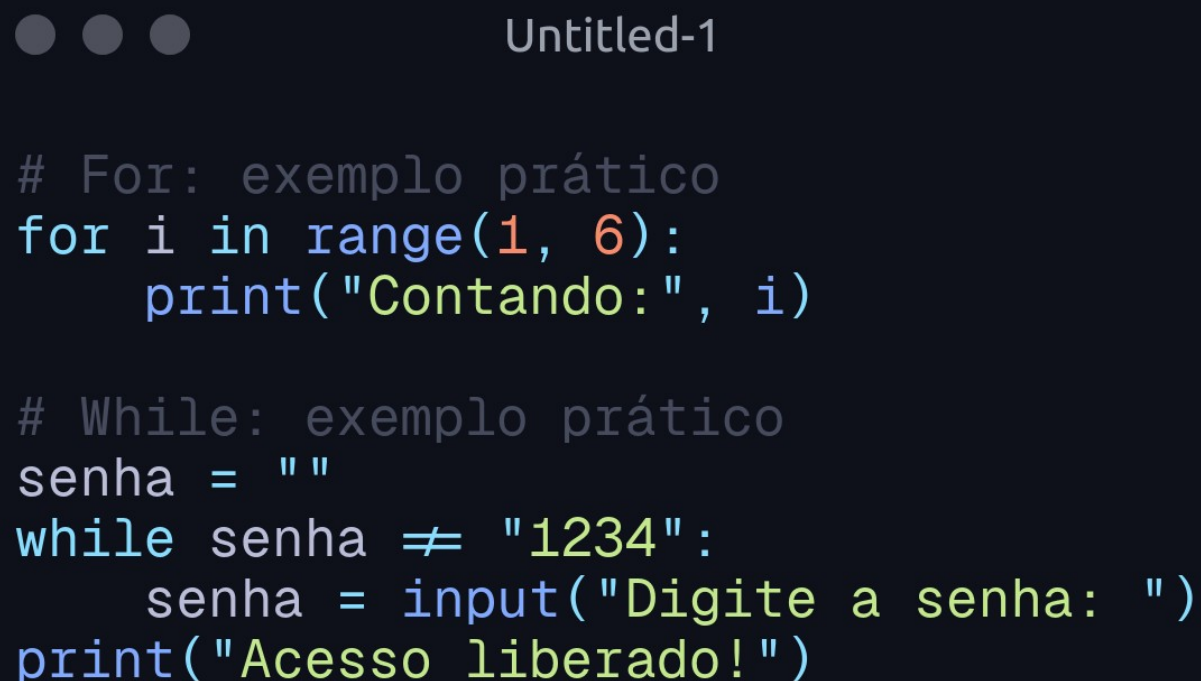
## Laços de Repetição

---

# Laços de Repetição: Automatizando Tarefas

## O que são laços?

Laços repetem comandos automaticamente. Use **`for`** quando souber quantas vezes repetir, e **“while”** para repetir até uma condição ser satisfeita.



```
● ● ● Untitled-1

# For: exemplo prático
for i in range(1, 6):
    print("Contando:", i)

# While: exemplo prático
senha = ""
while senha != "1234":
    senha = input("Digite a senha: ")
print("Acesso liberado!")
```

## Por que usar?

Ideal para processar listas, pedir tentativas de senha ou executar ações repetitivas, como enviar mensagens para vários contatos.

# 07

## **Estruturas de Dados**



# Estruturas de Dados: Guardando Vários Valores

## O que são listas?

Listas armazenam vários dados em uma única variável. Você pode acessar cada item pelo índice ou percorrer todos com laços.



Untitled-1

```
frutas = ["maçã", "banana", "uva"]  
for fruta in frutas:  
    print("Eu gosto de", fruta)
```

## Quando usar?

Use para guardar itens de compras, nomes de alunos, resultados de testes, etc.



08

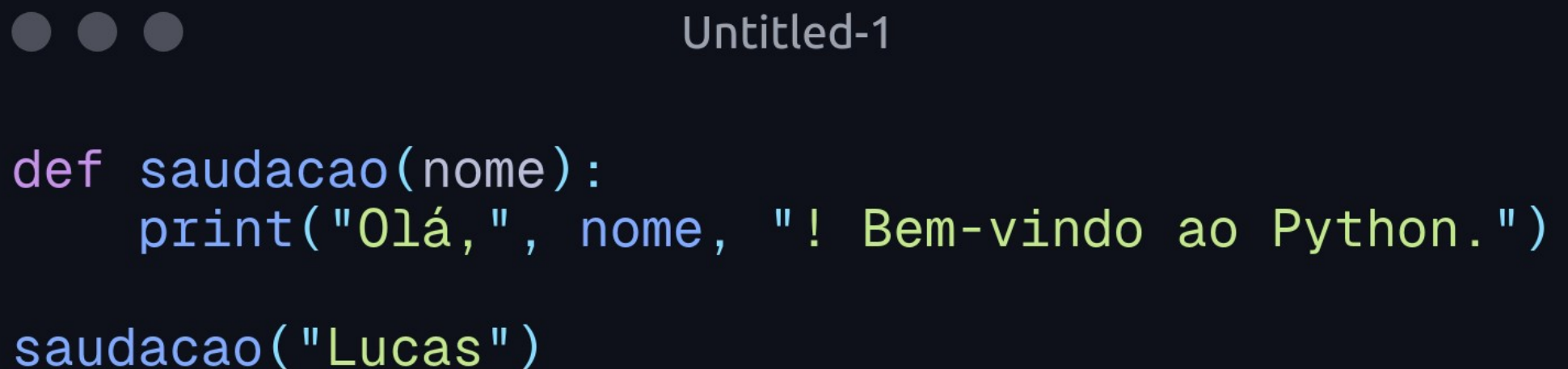
**Funções**



# Funções: Organize Seu Código

## Para que servem as funções?

Funções agrupam comandos que podem ser reutilizados em diferentes partes do programa. Isso evita repetição e deixa o código mais organizado.



```
def saudacao(nome):  
    print("Olá,", nome, "! Bem-vindo ao Python.")  
  
saudacao("Lucas")
```

## Vantagens

Facilita a manutenção e a leitura do código, além de permitir reaproveitamento em diferentes projetos.

# 09

## Comentários



# Comentários: Deixe o Código Claro

## O que são comentários?

Comentários são linhas ignoradas pelo Python, usadas para explicar o que cada parte do código faz. Começam com “#”.



Untitled-1

```
# Isso imprime uma mensagem de boas-vindas  
print("Bem-vindo ao curso de Python!")
```

## Por que usar?

Ajudam você e outras pessoas a entenderem o código no futuro, tornando a manutenção mais fácil.



# 10

## Erros Comuns



# Erros Comuns e Como Evitar

## Quais são os principais erros?

Os mais comuns são digitar algo errado (como esquecer aspas ou parênteses), usar tipos errados (tentar somar texto e número), ou esquecer de converter dados.



Untitled-1

```
# Erro: esqueci de fechar aspas  
print("Olá, mundo!)
```

## Como corrigir?

Leia a mensagem de erro com atenção e corrija a linha indicada na parte de problemas da sua IDE.

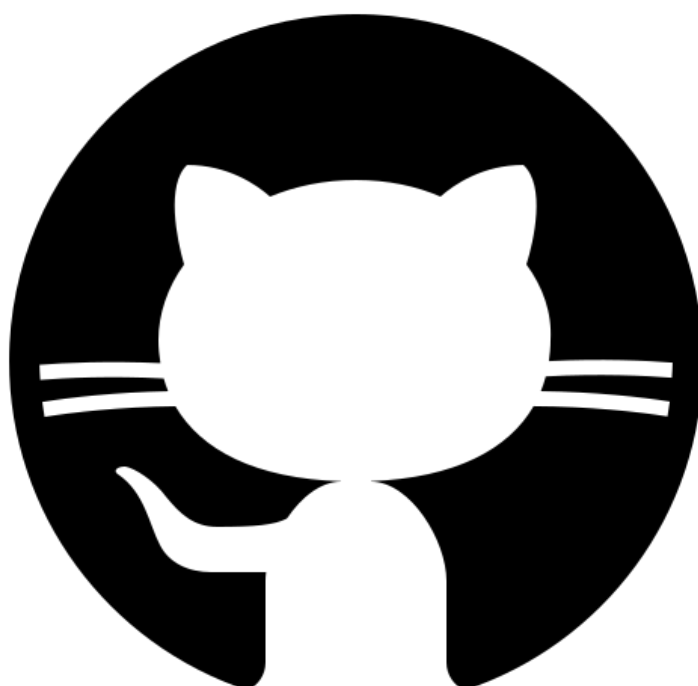
## Pratique!

A melhor forma de aprender é praticando. Experimente modificar os exemplos, crie pequenos projetos e explore as possibilidades do Python. A prática constante é o segredo do sucesso!

# Encerramento

Este E-Book foi criado com auxílio de inteligência artificial e faz parte de um estudo de prompts realizado no bootcamp da DIO (Digital Innovation One), no contexto do programa TONNIE Java and IA in Europe.

Espero que este material tenha sido útil para sua jornada de aprendizado em Python! Bons estudos e continue explorando novas possibilidades com tecnologia e IA.



[Clique aqui para ir até o repositório do e-book](#)