



# O GUIA DO PYTHON DAS GALAXIAS

Júlio César Senestro

# 01

# Introdução

---

Bem-vindo, explorador! Neste capítulo, vamos configurar nosso ambiente e dar os primeiros passos no mundo do Python.

# O que é Python?

Python é uma linguagem de programação poderosa e fácil de aprender. Ideal para quem está começando na programação.

## Instalando Python

1. Acesse o site oficial do Python: [python.org](https://python.org).
2. Baixe e instale a versão mais recente..

## Primeiro programa

Abra o editor de código ou o terminal e digite:

```
print("Olá, Universo!")
```

# 02

## Variáveis e tipos de dados

---

Para explorar novas galáxias, precisamos armazenar informações. Vamos aprender sobre variáveis e tipos de dados.

# Variáveis

Uma variável é um espaço de memória onde se pode armazenar dados.

Exemplo:

```
nome = "Aventureiro"  
idade = 21  
planeta = "Terra"  
  
print(nome)  
print(idade)  
print(planeta)
```

Criamos três variáveis: nome, idade, e planeta, cada uma armazenando um tipo diferente de informação.

# Tipo de dados

- String (str): Texto, por exemplo, "Olá"
- Inteiros (int): Números inteiros, por exemplo, 42
- Float (float): Números decimais, por exemplo, 3.14
- Booleano (bool): Verdadeiro ou Falso, por exemplo, True ou False

# 03

# Operações

---

Vamos aprender a fazer cálculos e tomar decisões com Python.

# Operações matemáticas

O Python pode realizar todas as operações matemáticas básicas como adição, subtração, multiplicação e divisão.

Exemplo:

```
a = 10
b = 5

soma = a + b
subtracao = a - b
multiplicacao = a * b
divisao = a / b
```

Neste exemplo, definimos duas variáveis a e b, e calculamos sua soma, subtração, multiplicação e divisão.



# Operações lógicas

As operações lógicas nos permitem comparar valores e tomar decisões baseadas nessas comparações.

Exemplo:

```
x = 10
y = 5

maior = x > y
menor = x < y
igual = x == y

print(maior, menor, igual)
```

Aqui, verificamos se x é maior que y, menor que y ou igual a y, e imprimimos os resultados que serão booleanos.

# 04

## Explorando loops e condicionais

---

Para viajar mais longe, precisamos repetir  
tarefas e tomar decisões inteligentes.

# Condicionais

As condicionais nos permitem executar diferentes blocos de código com base em certas condições. O uso do elif (abreviação de "else if") nos permite adicionar múltiplas condições.

Exemplo:

```
temperatura = 25

if temperatura > 30:
    print("Está muito quente!")
elif temperatura > 20:
    print("Está agradável.")
elif temperatura > 10:
    print("Está um pouco frio.")
else:
    print("Está muito frio!")
```

Se o valor de temperatura atender a uma das condições, o bloco em questão é executado.

# Loops

Os loops permitem repetir um bloco de código várias vezes

For loop

Exemplo:

```
for i in range(5):  
    print("Explorando o espaço:", i)
```

O for loop acima imprime "Explorando o espaço:" seguido do valor de i, que varia de 0 a 4.

## While loop

Exemplo:

```
contador = 0

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

O while loop continua executando o bloco de código enquanto a condição ( $\text{contador} < 5$ ) for verdadeira.

# 05

## Funções e módulos

---

Funções são nossas ferramentas para reutilizar código, e módulos são bibliotecas que nos ajudam a expandir nossas habilidades.

# Funções

Uma função é um bloco de código que só é executado quando é chamado. Você pode passar dados, conhecidos como parâmetros, para uma função.

Exemplo:

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

Aqui, definimos uma função `saudacao` que recebe um parâmetro `nome` e imprime uma mensagem de saudação.

# Módulos

Módulos são arquivos que contêm código Python e podem ser importados para outros programas.

Exemplo:

```
import math

raiz_quadrada = math.sqrt(16)
print(raiz_quadrada)
```

Usamos o módulo math para calcular a raiz quadrada de 16. Então, será mostrado 4.



# 06

## Listas e dicionários

---

Listas e dicionários são coleções que nos ajudam a organizar nossos dados.

# Listas

Uma lista é uma coleção ordenada de itens que podem ser alterados.

Exemplo:

```
planetas = ["Terra", "Marte", "Júpiter"]

print(planetas)
print(planetas[1])

planetas.append("Saturno")
print(planetas)
```

Aqui, definimos uma função `saudacao` que recebe um parâmetro `nome` e imprime uma mensagem de saudação.

# Dicionários

Um dicionário é uma coleção não ordenada de pares chave-valor.

Exemplo:

```
astro = {"nome": "Sol", "tipo": "estrela", "idade": 4.6}

print(astro)
print(astro["nome"])

astro["distancia"] = "149.6 milhões de km"
print(astro)
```

Criamos um dicionário para armazenar informações sobre o Sol e adicionamos um novo par chave-valor.

# Agradecimentos



# Sobre o ebook

Este ebook foi feito no bootcamp “Fundamentos de IA para Devs” da DIO utilizando ferramentas de Inteligência Artificial generativa, que incluem:

- ChatGPT: <https://chatgpt.com/>
- Microsoft Copilot: [copilot.Microsoft.com](https://copilot.microsoft.com)

O ebook aborda os conceitos mais elementares da linguagem Python. Ainda há um grande universo a percorrer na jornada de aprendizado. Então, prepare-se para a grande jornada!

Obrigado pela leitura”



# O GUIA DO PYTHON DAS GALAXIAS