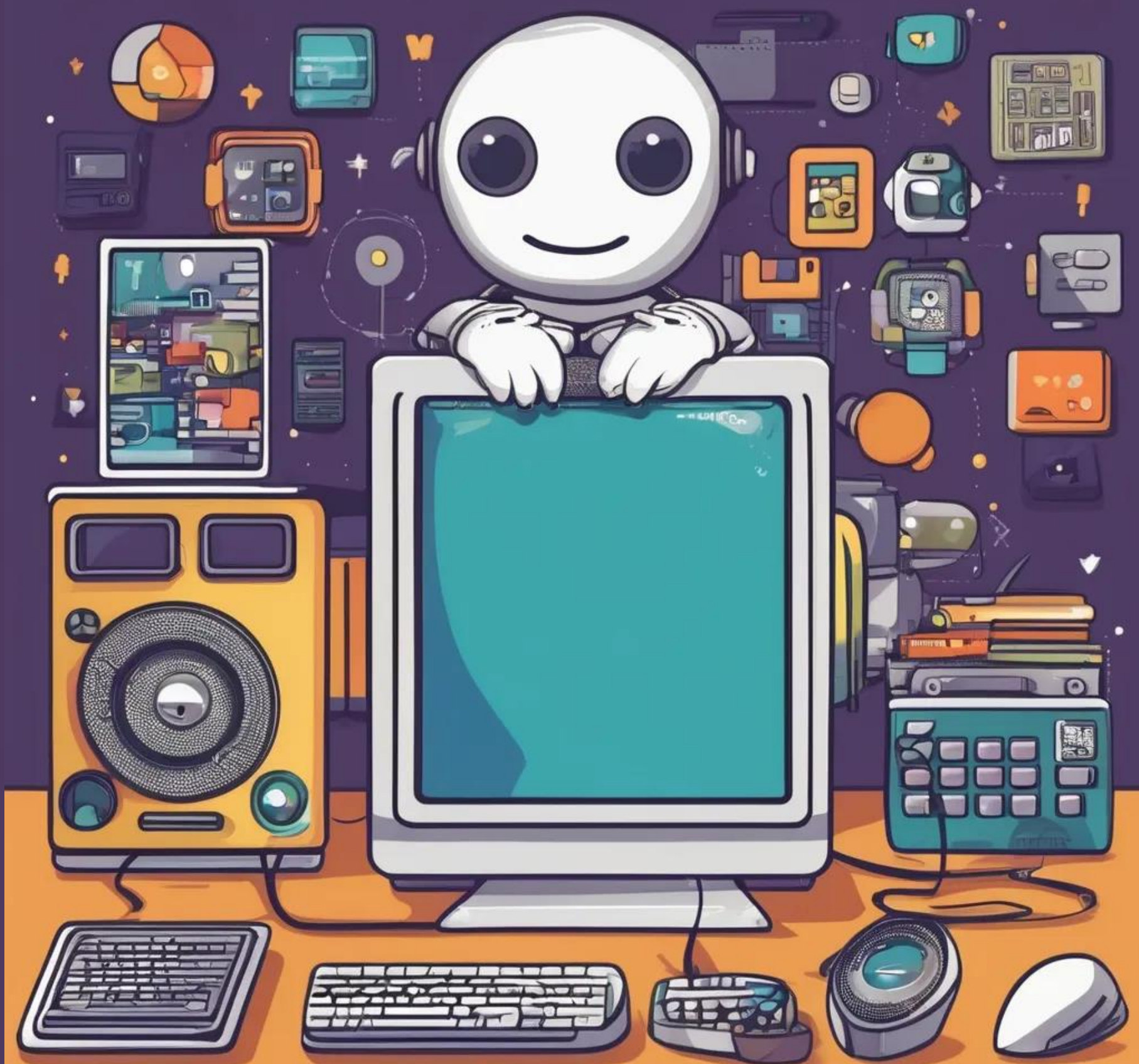


DESPERTAR DO PROGRAMADOR

O INÍCIO DA JORNADA



TABATA ETIÉLE

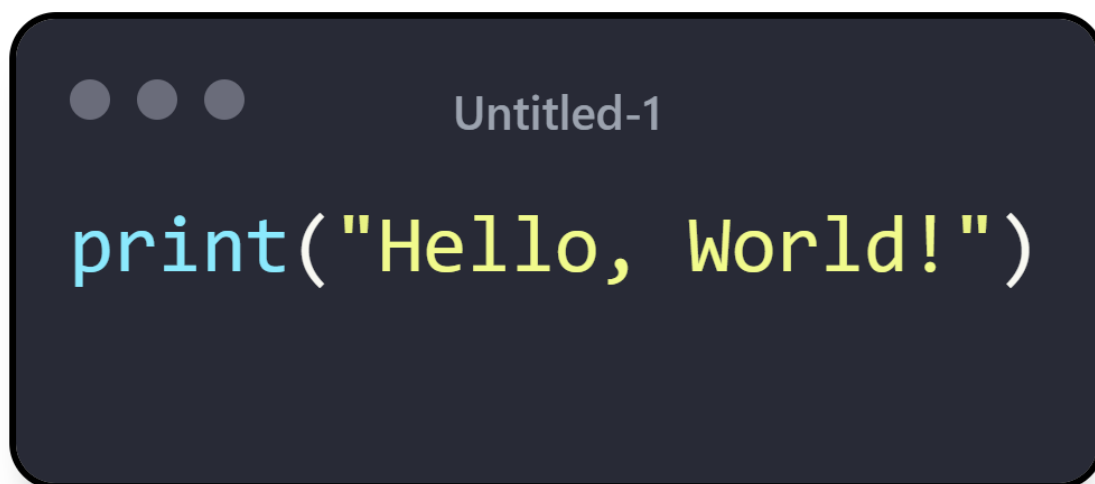
01

Fundamentos Básicos

OLÁ, MUNDO!

Comece com o básico:

Vamos começar com o básico: o programa "Hello, World!". Este é o primeiro programa que a maioria dos iniciantes aprende, pois ele introduz conceitos fundamentais como a sintaxe da linguagem e a saída de dados.

A dark-themed code editor window titled "Untitled-1" with three window control buttons (red, yellow, green) in the top-left corner. The code `print("Hello, World!")` is displayed in a light blue and yellow monospace font.

VARIÁVEIS

Guardando Informações

Variáveis são como caixas onde você guarda informações para usar mais tarde. Elas podem armazenar diferentes tipos de dados, como texto (strings), números inteiros e decimais.

```
variáveis  
  
nome = "Alice"  
idade = 25  
altura = 1.68
```



CONDICIONAIS

Fazendo Escolhas

As condicionais permitem que seu programa tome decisões com base em certas condições. Usamos if e else para executar diferentes blocos de código dependendo do valor das variáveis.

```
if/else

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



02

Estruturas de Controle e Funções

LAÇOS

Repetição Automática

Laços permitem repetir um bloco de código várias vezes. O laço for, por exemplo, é usado quando você sabe quantas vezes quer repetir uma ação.

```
laço  
  
for i in range(5):  
    print(f"Repetição {i}")
```



FUNÇÕES

Organizando o Código

Funções são blocos de código reutilizáveis que executam uma tarefa específica. Elas ajudam a organizar seu código e evitam a repetição.

```
funções

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

saudacao("Alice")
```



03

Estruturas de Dados e POO

LISTAS

Armazenando Coleções

Listas são usadas para armazenar múltiplos valores em uma única variável. Você pode acessar os itens de uma lista usando índices.

```
frutas = ["maçã", "banana", "cereja"]  
print(frutas[0]) # Saída: maçã
```



ENTRADA E SAÍDA DE DADOS

Dados Estruturados

Interagir com o usuário é uma parte fundamental da programação. Você pode usar `input()` para receber dados do usuário e `print()` para mostrar informações.



entrada e saída

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



MANIPULAÇÃO DE ARQUIVOS

Muitos programas precisam ler e escrever arquivos. Você pode usar a função `open()` para abrir um arquivo e métodos como `write()` e `read()` para escrever e ler dados.

```
arquivo

# Escrever em um arquivo
with open("arquivo.txt", "w") as file:
    file.write("Olá, Mundo!")

# Ler de um arquivo
with open("arquivo.txt", "r") as file:
    conteudo = file.read()
    print(conteudo)
```



PROGRAMAÇÃO ORIENTADA A OBJETOS (POO)

POO é um paradigma de programação que usa "objetos" – instâncias de classes – para modelar coisas do mundo real. Aqui está como definir uma classe Cachorro e criar um objeto dessa classe.

```
POO

class Cachorro:
    def __init__(self, nome, idade):
        self.nome = nome
        self.idade = idade

    def latir(self):
        print("Au Au!")

meu_cachorro = Cachorro("Rex", 2)
meu_cachorro.latir() # Saída: Au Au!
```



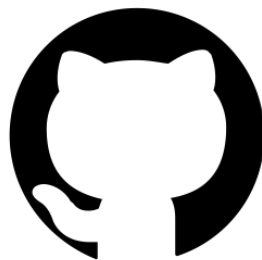
AGRADECIMENTOS



OBRIGADO POR LER

O conteúdo deste ebook foi criado por IA e editado pessoalmente.

Criado para fins de aprendizagem em curso para Dev's com fundamentos de IA.



LINK GITHUBE