PYTHON

O Guia Supremo da Arte Oculta

FRANKLIN SEIXAS

Guia Completo de Python para Programadores

Introdução

Python é uma linguagem poderosa e versátil, ideal para quem está começando no mundo da programação. Neste guia, abordaremos os principais conteúdos que você precisa dominar para programar em Python, acompanhados de exemplos práticos e reais.

01

INSTALANDO PYTHON

Configuração do Ambiente

Antes de começar a programar em Python, é necessário configurar o ambiente de desenvolvimento.

Instalando Python

Primeiro, baixe e instale a versão mais recente do Python a partir do site oficial python.org.

Configuração do Ambiente

Instalação de um IDE

Utilize um IDE (Ambiente de Desenvolvimento Integrado) como o PyCharm ou um editor de texto como o Visual Studio Code para facilitar o desenvolvimento.

02

SINTAXE BÁSICA

Hello World

O clássico exemplo "Hello World" é um bom ponto de partida para qualquer linguagem.

```
print("Hello, World!")
```

Variáveis e Tipos de Dados

Python é uma linguagem dinamicamente tipada, o que significa que você não precisa declarar o tipo de uma variável explicitamente.

```
nome = "Alice"
idade = 30
altura = 1.75
print(f"Nome: {nome}, Idade: {idade}, Altura: {altura}")
```

03

Estruturas de Controle

Condicionais

As estruturas condicionais permitem que seu código tome decisões baseadas em condições.

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

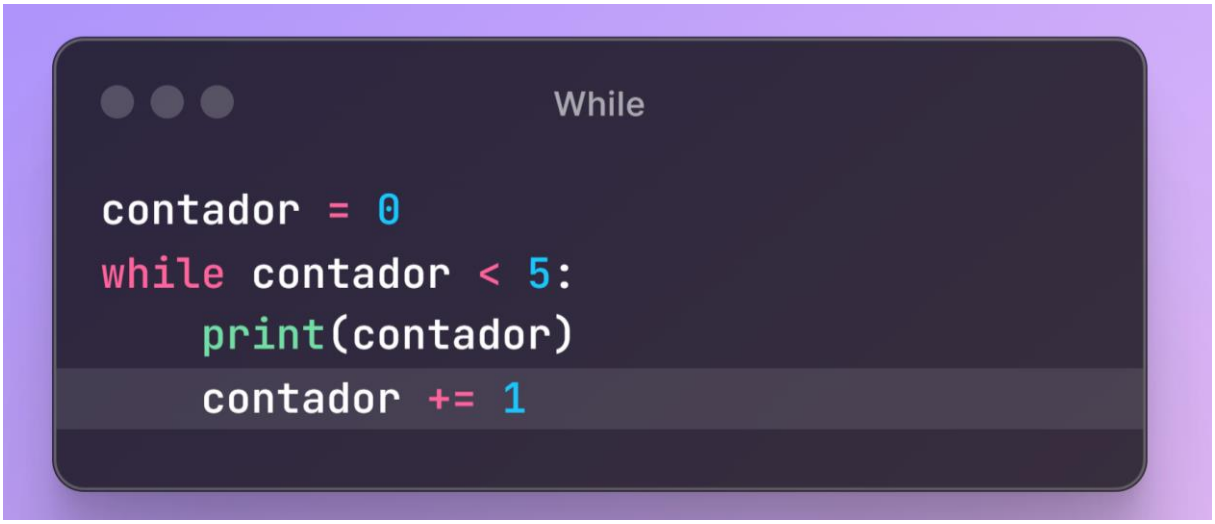
Laços de Repetição

Os laços permitem que você execute um bloco de código várias vezes.

For

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

While


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

04

Funções

Definindo Funções

Funções são blocos de código que realizam uma tarefa específica e podem ser reutilizadas.

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

print(saudacao("Carlos"))
```

Funções Lambda

As funções lambda são funções anônimas, úteis para operações simples.

```
quadrado = lambda x: x * x
print(quadrado(5))
```

Funções Lambda

As funções lambda são funções anônimas, úteis para operações simples.

```
quadrado = lambda x: x * x
print(quadrado(5))
```

05

Estruturas de Dados

Listas

Listas são coleções ordenadas de elementos.

```
numeros = [1, 2, 3, 4, 5]
print(numeros)
```

Dicionários

Dicionários armazenam pares chave-valor.

```
aluno = {"nome": "João", "idade": 21, "curso": "Engenharia"}
print(aluno["nome"])
```

Conjuntos

Dicionários armazenam pares chave-valor.

```
conjunto = {1, 2, 3, 4, 4, 5}
print(conjunto)  # Saída: {1, 2, 3, 4, 5}
```

06

Manipulação de Arquivos

Leitura de Arquivos

Você pode ler o conteúdo de arquivos com Python.

```
with open("arquivo.txt", "r") as arquivo:
    conteudo = arquivo.read()
    print(conteudo)
```

Escrita em Arquivos

Também é possível escrever em arquivos.

```
with open("saida.txt", "w") as arquivo:
    arquivo.write("Escrevendo em um arquivo com Python!")
```

Escrita em Arquivos

Também é possível escrever em arquivos.

```
with open("saida.txt", "w") as arquivo:
    arquivo.write("Escrevendo em um arquivo com Python!")
```

07

Programação Orientada a Objetos

Definindo Classes e Objetos

A POO (Programação Orientada a Objetos) é um paradigma que organiza o código em torno de objetos.

```python
class Pessoa:
    def __init__(self, nome, idade):
        self.nome = nome
        self.idade = idade

    def apresentar(self):
        return f"Meu nome é {self.nome} e eu tenho {self.idade} anos."

pessoa1 = Pessoa("Maria", 25)
print(pessoa1.apresentar())
```

08

Bibliotecas e Módulos

Importando Bibliotecas

Python possui diversas bibliotecas padrão e de terceiros que facilitam muitas tarefas.

```python
import math

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

Requests

Para fazer requisições HTTP.

```python
import requests

resposta = requests.get("https://api.github.com")
print(resposta.status_code)
```

09

Trabalhando com Dados

NumPy

NumPy é uma biblioteca poderosa para computação numérica.

```python
import numpy as np

array = np.array([1, 2, 3, 4])
print(array * 2)
```

Pandas

Pandas é excelente para manipulação e análise de dados.

```python
import pandas as pd

dados = {'Nome': ['Ana', 'João', 'Maria'], 'Idade': [28, 34, 29]}
df = pd.DataFrame(dados)
print(df)
```

10

Desenvolvimento Web

Flask

Flask é um microframework para desenvolvimento web.

```python
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Hello, Flask!"

if __name__ == '__main__':
    app.run(debug=True)
```

Django

Django é um framework completo para desenvolvimento web.

```python
# Exemplo simplificado de configuração de um projeto Django
# Este exemplo assume que você já tem Django instalado e um projeto criado.

# No arquivo views.py
from django.http import HttpResponse

def hello_world(request):
    return HttpResponse("Hello, Django!")
```

CONCLUSÕES

Este eBook apresentou os principais conteúdos para você começar a programar em Python, com exemplos práticos em contextos reais. Python é uma linguagem rica e versátil, e dominar seus conceitos básicos abrirá portas para diversas áreas da tecnologia. Continue praticando e explorando as diversas bibliotecas e frameworks disponíveis, e você estará preparado para enfrentar qualquer desafio no mundo da programação.