


Python na Prática: *Fundamentos* *Essenciais para Iniciantes*



Prof. Me. Diego H. Negretto



Python na **Prática:** ***Fundamentos*** ***Essenciais*** ***para Iniciantes***

Objetivos:

Capacitar os participantes a entender e aplicar conceitos essenciais da linguagem de programação Python.

Público Alvo:

Estudantes e profissionais interessados em programação, sem necessidade de experiência prévia.

Carga horária:

12 horas - semanal, aos sábados, das 08h15 às 12h15

Início e Término:

30 de agosto a 13 de setembro de 2025



Python na **Prática:** ***Fundamentos*** ***Essenciais*** ***para Iniciantes***

Conteúdo programático

Encontro 01: Fundamentos do Python (4h)

- Visão geral do Python: História, aplicações e por que aprender Python;
- Configuração do ambiente: Instalação do Python, VS Code, Jupyter Notebook;
- Estrutura de um programa em Python;
- Variáveis, tipos de dados e operadores;
- Entrada e saída de dados (input() e print());
- Introdução a estruturas condicionais (if, elif, else);
- Estruturas de repetição: for e while;
- Mão na massa: Exercícios práticos.



Python na **Prática:** ***Fundamentos*** ***Essenciais*** ***para Iniciantes***

Conteúdo programático

Encontro 02: Funções e Manipulação de Arquivos (4h)

- Listas e tuplas: Manipulação de coleções de dados;
- Introdução a dicionários e conjuntos;
- Funções: Definição, parâmetros e retorno;
- Manipulação de arquivos (open, leitura e escrita);
- Introdução ao tratamento de exceções (*try*, *except*, *finally*);
- Boas práticas na organização do código;
- Mão na massa: Exercícios práticos.



Python na **Prática:** ***Fundamentos*** ***Essenciais*** ***para Iniciantes***

Conteúdo programático

Encontro 03: Introdução a Bibliotecas e Mini-projeto (4h)

- Uso de bibliotecas padrão do Python (*os*, *math*, *datetime*);
- Introdução ao *pandas* e *matplotlib* para manipulação de dados e visualização;
- Mão na massa: Desenvolvimento de um mini-projeto (Aplicação simples para análise de dados);
- Encerramento e discussão de próximas etapas para aprofundamento em Python.



Introdução ao Python

Bibliotecas padrão do Python

As **bibliotecas padrão** do Python são um **conjunto de módulos e pacotes de código que já vêm instalados com a linguagem**. A grande vantagem é que você não precisa instalar nada extra para começar a usá-las; elas estão sempre disponíveis.



Introdução ao Python

Bibliotecas padrão do Python

Principais Bibliotecas Padrão do Python

- **os:** Este módulo permite interagir com o sistema operacional.
- **sys:** Fornece acesso a variáveis e funções relacionadas ao interpretador Python.
- **math:** Contém funções matemáticas essenciais.
- **datetime:** Essencial para trabalhar com datas e horas.
- **json:** Ideal para trabalhar com dados no formato JSON (JavaScript Object Notation).
- **random:** Usado para gerar números aleatórios.
- **re:** Fornece suporte para expressões regulares.
- **sqlite3:** Um módulo para interagir com bancos de dados SQLite.



Introdução ao Python

Bibliotecas externas (*Third-party*)

Bibliotecas externas, também conhecidas como bibliotecas de terceiros (third-party), são módulos de código que não vêm instalados com o Python. Diferente das bibliotecas padrão, elas são criadas e mantidas pela comunidade de desenvolvedores para resolver problemas específicos.

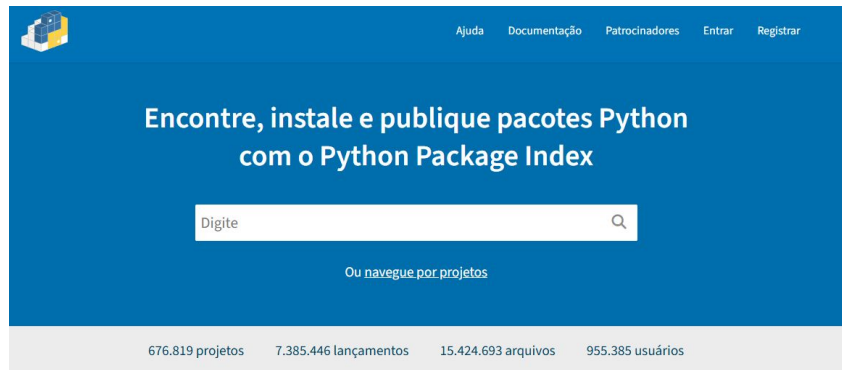


Introdução ao Python

Gerenciador de pacotes (pip)

O **pip** é o sistema de gerenciamento de pacotes padrão do Python. Sua principal função é se conectar ao PyPI (*Python Package Index*), que é o repositório oficial da comunidade para pacotes Python.

<https://pypi.org/>





Introdução ao Python

Gerenciador de pacotes (pip)

Utilização básica:

- **Instalar pacotes:** Você executa *pip install nome_do_pacote* para baixar e instalar uma biblioteca no seu sistema.
- **Desinstalar pacotes:** Com o comando *pip uninstall nome_do_pacote*, você remove um pacote que não precisa mais.
- **Gerenciar dependências:** O pip também lida com as dependências, garantindo que, quando você instala um pacote, todos os outros pacotes de que ele precisa também sejam instalados automaticamente.



Introdução ao Python

Ambientes Virtuais em Python

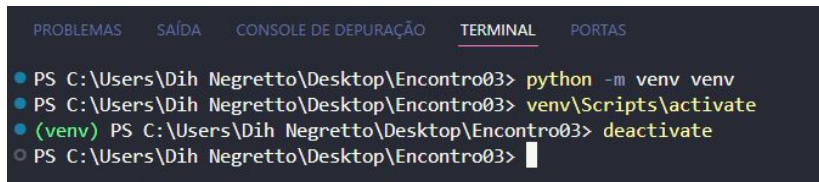
Um ambiente virtual em Python é um diretório isolado que contém sua própria cópia do interpretador Python e de todas as bibliotecas que você instala para um projeto específico.



Introdução ao Python

Ambientes Virtuais em Python

Criação, ativação e desativação no Windows:



```
PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL  PORTAS
● PS C:\Users\Dih Negretto\Desktop\Encontro03> python -m venv venv
● PS C:\Users\Dih Negretto\Desktop\Encontro03> venv\Scripts\activate
● (venv) PS C:\Users\Dih Negretto\Desktop\Encontro03> deactivate
○ PS C:\Users\Dih Negretto\Desktop\Encontro03> 
```

- O primeiro comando é para criar o ambiente virtual. O comando básico é ***python -m venv nome_ambiente***
- O **nome_ambiente** pode ser qualquer coisa, mas é comum usar **venv** ou **.venv** para facilitar a identificação.
- O segundo comando é para ativar o ambiente virtual.
- Para ativar o ambiente no macOS e Linux o comando é **source nome_ambiente/bin/activate**
- O terceiro comando desativa o ambiente virtual.



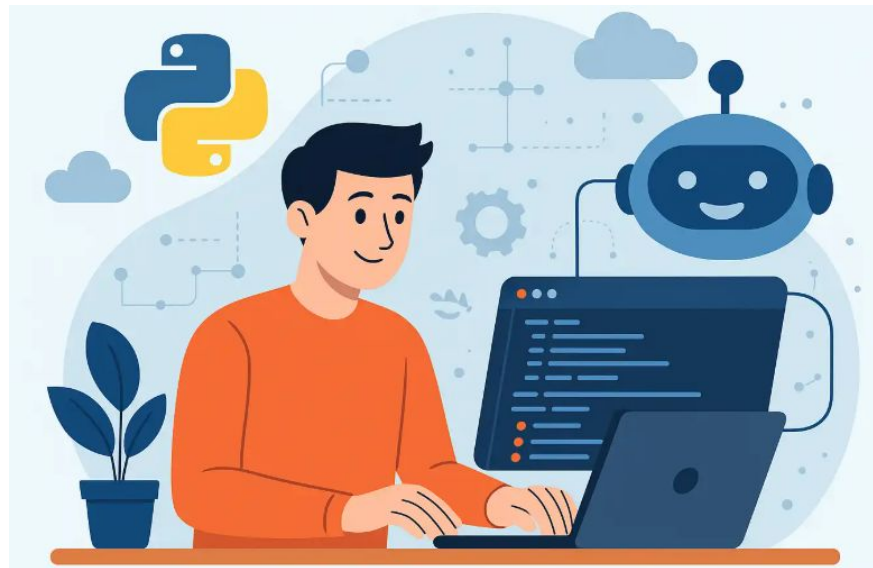
Introdução ao Python

Importação de bibliotecas

```
bibliotecas.py ×  
bibliotecas.py  
1  # Importação completa  
2  import math  
3  
4  # Importação com alias (apelido)  
5  import pandas as pd  
6  
7  # Importação seletiva de partes específicas  
8  from math import sqrt  
9  
10 # Importação seletiva de várias partes  
11 from math import sqrt, pow  
12  
13 # Importação de Todos os Conteúdos (Não Recomendada)  
14 from math import *
```

Introdução ao Python

Mão na Massa:
Aplicação simples para
análise de dados





Introdução ao Python

Próximos passos

Desenvolvimento Web com Python:

- Fundamentos da Web: HTTP, REST, JSON, HTML, CSS, JavaScript.
- Frameworks:
 - Django (para aplicações completas e robustas).
 - Flask ou FastAPI (para APIs mais leves).
- Banco de Dados: SQL (PostgreSQL/MySQL) + ORM (Django ORM, SQLAlchemy).
- Boas práticas: testes, autenticação/autorização, segurança.
- Deploy: Docker, CI/CD, hospedagem em nuvem (AWS, GCP, Heroku, etc).



Introdução ao Python

Próximos passos

DataScience com Python:

- Fundamentos de Estatística e Probabilidade.
- Bibliotecas essenciais: Pandas, NumPy, Matplotlib, Seaborn.
- Exploração e limpeza de dados (ETL).
- Machine Learning aplicado: Scikit-learn.
- Visualização avançada: Plotly, Dash.
- Big Data e Cloud: Spark, Hadoop, Google BigQuery.
- Comunicação de resultados: Jupyter Notebook, relatórios, storytelling com dados.



Introdução ao Python

Próximos passos

Inteligência Artificial e Python:

- Conceitos de IA e ML: aprendizado supervisionado e não supervisionado.
- Redes Neurais e Deep Learning (PyTorch ou TensorFlow).
- LangChain: para orquestração de LLMs, memória e ferramentas.
- LlamaIndex: para conectar agentes a dados externos (RAG).
- OpenAI API / Hugging Face Inference: integração com modelos prontos.