

Conteúdo Programático:

- Curso autodirigido -

ABC do Python

Por: Vicente Freire



Fevereiro

Aula 1: Introdução à Programação e Primeiros Passos

Neste tópico, você dará seus primeiros passos no mundo da programação! Vamos entender o que é programação, instalar o Python e escolher uma IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado que é um software que reúne várias ferramentas para facilitar a programação) afim de escrever nosso código. Além disso, você aprenderá a rodar seus primeiros comandos e a entender a sintaxe básica da linguagem.

Aula 2: Tipos de Dados e Operações Básicas

Neste módulo, você será apresentado aos principais tipos de dados utilizados em Python, como inteiros (int), números de ponto flutuante (float), strings (str) e booleanos (bool), aprendendo a armazenar e manipular diferentes tipos de informações. Você também explorará operadores matemáticos, como soma, subtração, multiplicação e divisão, além de aprender como realizar conversões entre tipos de dados para garantir que seu código manipule dados de maneira correta e eficiente.



Fevereiro

Aula 3: Estruturas Condicionais e Operadores Lógicos: As estruturas condicionais são fundamentais para a criação de fluxos de decisão em um programa. Neste módulo, você aprenderá a utilizar as instruções if, elif e else para implementar lógica condicional. Além disso, explorará os operadores lógicos (and, or, not), essenciais para a construção de expressões booleanas mais complexas.

Aula 4: Laços de Repetição e Controle de Fluxo: Os laços de repetição permitem a execução iterativa de um bloco de código, tornando o programa mais eficiente e reduzindo a necessidade de repetições manuais. Você aprenderá a utilizar os loops for e while, bem como comandos de controle de fluxo (break, continue) para gerenciar a execução dos laços conforme necessário.

Aula 5: Manipulação de Listas : As listas são uma das estruturas de dados mais utilizadas em Python devido à sua flexibilidade. Neste tópico, você aprenderá a criar e modificar listas, realizar inserções e remoções de elementos e iterar sobre seus itens utilizando laços de repetição. Também explorará métodos essenciais como append(), remove() e pop().



Fevereiro

Aula 6: Introdução a Tuplas e Dicionários: Além das listas, Python oferece outras estruturas de dados poderosas. As tuplas são sequências imutáveis utilizadas para armazenar coleções de dados de forma eficiente. Já os dicionários permitem a associação de chaves a valores, sendo úteis para armazenar informações estruturadas de maneira organizada. Você aprenderá como criar, acessar e manipular essas estruturas no seu código.

Aula 7: Funções e Escopo de Variáveis: As funções permitem a modularização do código, tornando-o mais organizado e reutilizável. Neste tópico, você aprenderá a definir funções com parâmetros e valores de retorno, além de compreender a diferença entre escopo local e global de variáveis. Também será introduzida a construção de funções anônimas utilizando lambda, que possibilita a criação de expressões mais concisas.

Aula 8: Manipulação de Arquivos : A manipulação de arquivos é essencial para leitura e armazenamento de dados em programas. Você aprenderá a abrir, ler e escrever arquivos .txt e .csv utilizando a função `open()` e a biblioteca `csv`. Esse conhecimento é fundamental para o processamento de grandes volumes de dados e para a criação de sistemas que interagem com arquivos externos.



Março

Aula 9: Introdução ao Tratamento de Erros e Exceções

Erros e exceções fazem parte do desenvolvimento de software, e saber tratá-los adequadamente evita falhas inesperadas nos programas. Neste módulo, você aprenderá a identificar diferentes tipos de erros e a utilizar blocos try, except, else e finally para lidar com exceções de forma segura e eficiente.

Aula 10: Módulos e Bibliotecas Nativas do Python: A linguagem Python possui uma vasta biblioteca padrão com módulos que facilitam diversas tarefas. Você aprenderá a utilizar o módulo “os” para manipulação do sistema operacional, “math” para operações matemáticas avançadas e “datetime” para trabalhar com datas e horários.

Aula 11: Sets e Estruturas de Dados Especiais: Além de listas, tuplas e dicionários, Python oferece outras estruturas de dados como set, que permite armazenar conjuntos de elementos únicos. Neste módulo, você aprenderá como criar, manipular e aplicar conjuntos e outras estruturas especiais para otimizar o processamento de dados.



Março

Aula 12: Introdução à Programação Orientada a Objetos: Classes e Objetos : A Programação Orientada a Objetos (POO) permite estruturar programas de forma modular e reutilizável. Aqui, você aprenderá os conceitos fundamentais de POO, como a definição de classes e a criação de objetos, entendendo como modelar entidades do mundo real em código.

Aula 13: Métodos, Dunder Methods e Atributos: Neste tópico, exploraremos os métodos dentro das classes, diferenciando métodos de instância, de classe e estáticos. Além disso, você aprenderá sobre dunder methods (métodos mágicos), como `__init__` e `__str__`, que permitem personalizar o comportamento dos objetos.

Aula 14: Herança, Polimorfismo e Encapsulamento: A reutilização de código é uma das principais vantagens da POO. Você aprenderá a aplicar herança, permitindo que classes herdem características de outras, além de entender polimorfismo, que possibilita a sobrescrita de métodos. O conceito de encapsulamento também será abordado, garantindo maior controle sobre o acesso a atributos e métodos.



Março

Aula 15: Relações entre Classes: Associação, Agregação e Composição

Para modelar sistemas complexos, as classes podem se relacionar de diferentes formas. Você aprenderá os conceitos de **associação** (quando uma classe usa outra), **agregação** (quando uma classe contém outra, mas sem dependência direta) e **composição** (quando uma classe depende totalmente da outra para existir).

Aula 16: POO Aplicada a Bibliotecas Externas, virtualenv e pip

Além da biblioteca padrão do Python, existem diversas bibliotecas externas que podem ser utilizadas para diferentes finalidades. Neste módulo, você aprenderá a instalar e gerenciar pacotes com o pip, criar ambientes virtuais (virtualenv) e utilizar POO para integrar bibliotecas externas aos seus projetos.



Abril

Aula 17: Introdução à Automação de Tarefas e Ferramentas Comuns :Automatizar tarefas repetitivas melhora a produtividade no desenvolvimento de software. Aqui, você aprenderá como Python pode ser utilizado para automatizar processos do dia a dia, desde manipulação de arquivos até interações com a web e sistemas operacionais.

Aula 18: Automatizando Tarefas com Scripts: Escrever scripts eficientes permite automatizar tarefas administrativas, como movimentação de arquivos, backup de dados e execução de comandos em sistemas operacionais. Você aprenderá a escrever scripts Python e Bash para automação de fluxos de trabalho.

Aula 19: . Introdução ao Git e GitHub: Conceitos e Fluxo de Trabalho: O controle de versão é essencial para qualquer desenvolvedor. Neste módulo, você aprenderá os fundamentos do **Git**, entendendo como versionar seu código e colaborar em projetos. Além disso, explorará o **GitHub**, aprendendo a criar repositórios, clonar projetos e contribuir com código aberto.



Abril

Aula 20: Trabalhando com Branches e Git Flow : O uso de branches permite trabalhar em diferentes versões do código simultaneamente. Aqui, você aprenderá a criar e gerenciar branches, além de entender o conceito de Git Flow, um fluxo de trabalho que ajuda a organizar o desenvolvimento de software de forma eficiente.

Aula 21: Automatizando Testes com GitHub Actions : Os testes automatizados garantem a qualidade do código e evitam erros em produção. Neste módulo, você aprenderá a configurar GitHub Actions para rodar testes automaticamente sempre que houver uma atualização no repositório, garantindo um fluxo de desenvolvimento mais seguro e eficiente.

Aula 22: Criando e Organizando o Seu Portfólio : Ter um portfólio bem estruturado é essencial para destacar seus projetos e habilidades. Você aprenderá a criar um repositório no GitHub, organizar seus projetos, escrever descrições eficazes e tornar seu portfólio mais atrativo para recrutadores e empresas.



Abril

Aula 23: Documentando Projetos com Markdown e GitHub

Wiki : A documentação clara e bem escrita facilita a manutenção e a colaboração em projetos. Neste módulo, você aprenderá a usar Markdown para escrever arquivos README.md, criar documentações organizadas com o GitHub Wiki e adicionar exemplos e guias para seus projetos.

Aula 24: Melhorando o Workflow e Colaboração com Pull

Requests: O fluxo de trabalho colaborativo é essencial em projetos compartilhados. Você aprenderá a criar e revisar Pull Requests, sugerindo e incorporando melhorias ao código de forma organizada. Além disso, verá como utilizar revisões de código e discussões para um desenvolvimento mais eficiente em equipe.



Maio

Aula 25: Introdução ao Pandas: Estruturas de Dados e Operações Básicas: O Pandas é uma das bibliotecas mais utilizadas para análise de dados em Python. Aqui, você aprenderá a trabalhar com DataFrames e Series, explorando operações básicas para manipulação e transformação de dados tabulares.

Aula 26: Introdução ao NumPy: Arrays e Operações : O **NumPy** é uma biblioteca fundamental para cálculos numéricos e manipulação de arrays multidimensionais. Você aprenderá a criar arrays, realizar operações matemáticas vetorizadas e utilizar funções eficientes para processamento de grandes volumes de dados.

Aula 27: Leitura e Manipulação de Arquivos Excel e CSV com Pandas: Neste módulo, você aprenderá a ler, manipular e salvar arquivos nos formatos CSV e Excel utilizando o Pandas. Verá como importar grandes conjuntos de dados, realizar transformações e exportar resultados de forma otimizada.



Maio

Aula 28: Visualização de Dados com Matplotlib: A visualização de dados é essencial para a análise exploratória e apresentação de insights. Você aprenderá a utilizar o Matplotlib para criar gráficos simples, como linhas, barras e dispersão, tornando seus dados mais compreensíveis.

Aula 29: Limpeza e Preparação de Dados no Pandas

Dados do mundo real frequentemente possuem inconsistências. Aqui, você aprenderá técnicas de limpeza e pré-processamento de dados, incluindo remoção de valores ausentes, tratamento de duplicatas e conversão de tipos de dados no Pandas.

Aula 30: Introdução à Estatística Descritiva com Python

A estatística descritiva é fundamental para entender padrões e tendências nos dados. Você aprenderá a calcular médias, medianas, desvios-padrão, distribuições e correlações utilizando Pandas e outras bibliotecas estatísticas.



Maio

Aula 31: Gráficos Avançados: Neste módulo, você irá aprofundar seus conhecimentos em visualização de dados, criando gráficos mais sofisticados, incluindo heatmaps, gráficos de densidade e histogramas, personalizando estilos e aprimorando a apresentação dos dados.

Aula 32: Introdução à Inteligência Artificial e Machine Learning: Este módulo apresenta os conceitos fundamentais de Inteligência Artificial (IA) e Aprendizado de Máquina (ML). Você aprenderá como os algoritmos de ML funcionam, explorando modelos básicos e compreendendo como treinar e avaliar modelos preditivos utilizando bibliotecas como Scikit-Learn.



Junho

Aula 33: Fundamentos de APIs e Requisições HTTP com requests: As APIs são fundamentais para a comunicação entre sistemas e a troca de dados na web. Neste módulo, você aprenderá os conceitos essenciais de APIs, como fazer requisições HTTP usando a biblioteca requests. Você será capaz de enviar requisições GET, POST, PUT e DELETE, manipular os dados recebidos e integrar essas informações em seus projetos.

Aula 34 : Introdução ao Flask: Criação de Rotas Básicas : Flask é um microframework de Python para o desenvolvimento de aplicações web. Aqui, você aprenderá a criar uma aplicação web simples utilizando o Flask, incluindo a configuração de rotas e visualizações. Você entenderá como mapear URLs para funções específicas e responder com conteúdo dinâmico para o usuário.

Aula 35 : Introdução ao SQLite e Conceitos Básicos de Banco de Dados : O SQLite é um sistema de banco de dados leve e amplamente utilizado para armazenamento local. Neste módulo, você aprenderá o conceito de banco de dados relacionais, como tabelas, linhas e colunas, e entenderá a linguagem SQL para realizar operações básicas de inserção, consulta, atualização e exclusão de dados.



Junho

Aula 36: Integração de Python com SQLite

Python permite integrar de forma eficiente com bancos de dados SQLite. Neste módulo, você aprenderá a **conectar** seu código Python a um banco de dados SQLite, realizar **consultas SQL** diretamente do Python e manipular os dados de forma dinâmica, garantindo a persistência de informações dentro de suas aplicações.

Aula 37 : Integração de Dados em Tempo Real com Flask e Pandas

A combinação de **Flask** com **Pandas** permite criar aplicações web dinâmicas que manipulam dados em tempo real. Neste módulo, você aprenderá a integrar essas duas ferramentas para ler, processar e exibir dados atualizados diretamente no navegador, criando dashboards interativos e eficientes.

Aula 38 : Estruturação de APIs Completas com Flask

Neste módulo, você aprenderá a estruturar **APIs RESTful** completas utilizando Flask. A partir da criação de rotas e definição de métodos HTTP, você será capaz de integrar sua API a bancos de dados, tratar erros de forma adequada e estruturar sua aplicação de maneira modular, permitindo que ela seja escalável e fácil de manter.



Junho

Aula 39: Conceitos Básicos de Web Scraping com

BeautifulSoup : Web scraping é uma técnica usada para extrair dados de websites. Neste módulo, você aprenderá a utilizar a biblioteca BeautifulSoup para parsing de páginas HTML, localizar e extrair informações relevantes como texto, links e imagens, além de entender as melhores práticas para coletar dados de forma ética e legal.

Aula 40: Introdução ao Django: Visão Geral e Estrutura Básica

: O Django é um framework robusto para o desenvolvimento de aplicações web em Python. Neste módulo, você será introduzido à estrutura MVC do Django, aprenderá sobre o modelo de rotas e views, além de configurar seu ambiente de desenvolvimento. Ao final, você terá a base para criar aplicações web completas com Django de forma eficiente.



Julho

Aula 41: Introdução à Interface Gráfica: Interfaces gráficas (GUI) permitem criar aplicações interativas com botões, janelas e formulários. Neste módulo, você será introduzido ao desenvolvimento de interfaces gráficas em Python utilizando bibliotecas como Tkinter, PySide ou PySimpleGUI, aprendendo a criar sua primeira janela interativa.

Aula 42: Widgets e Layouts : Os widgets são os elementos gráficos de uma interface, como botões, caixas de texto e menus. Aqui, você aprenderá a utilizar diferentes widgets e a organizar sua interface de maneira eficiente com layouts responsivos, garantindo melhor usabilidade.

Aula 43: Manipulação de Eventos : As aplicações gráficas precisam reagir a interações do usuário, como cliques e digitação. Neste módulo, você aprenderá a manipular eventos e sinais para capturar ações do usuário e executar funções específicas em resposta a essas interações.



Julho

Aula 44: Trabalhando com Widgets Avançados: Além dos widgets básicos, existem elementos gráficos mais complexos, como barras de progresso, menus suspensos e tabelas. Aqui, você aprenderá a personalizar e integrar widgets avançados para criar interfaces mais dinâmicas e completas.

Aula 45: Gerenciamento de Janela e Personalização: Este módulo explora técnicas para personalizar janelas e melhorar a experiência do usuário. Você aprenderá a ajustar dimensões, alterar estilos visuais, criar menus de navegação e gerenciar múltiplas janelas dentro da aplicação.

Aula 46: Integração com Arquivos e Banco de Dados: Aplicações reais geralmente precisam armazenar e recuperar dados. Neste módulo, você aprenderá a salvar e carregar informações de arquivos e a integrar sua interface gráfica com bancos de dados, utilizando SQLite para armazenar e gerenciar registros de forma eficiente.



Julho

Aula 47: Aplicações Complexas e Multi-threading: À medida que as aplicações se tornam mais avançadas, podem surgir problemas de desempenho. Aqui, você aprenderá a utilizar multi-threading para executar múltiplas tarefas simultaneamente, garantindo interfaces responsivas mesmo durante operações demoradas.

Aula 48: Empacotamento e Distribuição: Após desenvolver sua aplicação, é importante distribuí-la para outros usuários. Neste módulo, você aprenderá a empacotar seu programa em um executável, tornando-o independente do Python, e a criar instaladores utilizando ferramentas como PyInstaller e cx_Freeze.

**Muito obrigado
por ter lido até aqui!**

**Espero ter você como aluno(a)
Bora aprender Python comigo?**

Inscreva-se aqui:

Link para inscrição:

<https://abcdopython.com.br/login/index.php>