

12-WEEK PROGRAM

# Python for Programming

A Practical, Industry-Aligned 3-Month Course

Master Python from the ground up — through hands-on projects, live mentoring, and real-world industry applications. This course is built for early professionals, graduate students, and career-transitioners who want job-ready Python skills fast.

|                                                                                  |                                                                                  |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| <div>3 Months</div> <div>12 structured weeks of learning</div>                   | <div>Blended Format</div> <div>Recorded sessions + live 1:1 mentoring</div>      |
| <div>Industry-First</div> <div>70–80% core concepts, 20–30% real use cases</div> | <div>Job-Ready</div> <div>Portfolio projects &amp; skills mapping included</div> |

## WHY THIS COURSE

# Why Python? Why Now?

Python has become the most in-demand programming language across industries — from finance and healthcare to tech and e-commerce. Whether you're building automation tools, analyzing data, or deploying AI models, Python is the common thread.

## The Market Reality

- Python ranks #1 on the TIOBE Index for 3+ consecutive years
- Over 70% of data science and ML job postings require Python
- Avg. Python developer salary: \$110K–\$140K in the U.S.
- Used by Google, Netflix, NASA, JPMorgan, and thousands of startups

## What This Course Solves

- Most learners know syntax but can't apply Python to real problems
- Self-taught paths lack structure and mentorship
- Academic courses focus on theory, not industry workflows
- This course bridges the gap — concept mastery + practical execution

## COURSE OVERVIEW

# Tools & Technology Stack

You'll work with the same tools professionals use on the job — from local development environments to cloud notebooks and version control platforms.



## Core Language

Python 3.x — the current industry standard. No legacy syntax; fully modern and compatible with all major frameworks.



## Libraries & Frameworks

NumPy, Pandas, Matplotlib, Seaborn, Flask, SQLite3, Requests — progressively introduced across modules.



## Development Environments

VS Code, PyCharm, Jupyter Notebook, and Google Colab — covering local and cloud-based workflows.



## Collaboration & Versioning

Git & GitHub for version control, code sharing, and building a visible portfolio for job applications.

COURSE ROADMAP

# 12-Week Learning Roadmap

The course is structured in three progressive phases — each building on the last. You move from fundamentals to real-world application, with increasing complexity and independence.

1

## Phase 1 — Foundation

### Weeks 1–4

- Python syntax & setup
- Data types & control flow
- Functions & modules
- File I/O & error handling

2

## Phase 2 — Intermediate

### Weeks 5–8

- OOP principles
- Data manipulation (Pandas)
- APIs & web scraping
- Database integration

3

## Phase 3 — Advanced

### Weeks 9–12

- Automation & scripting
- Data visualization
- Intro to ML pipelines
- Capstone project

# Foundation Phase: Weeks 1-4

Build a rock-solid Python foundation. These four weeks ensure you're comfortable with Python's core building blocks before tackling more complex concepts.

## 1

### Week 1 — Setup & Python Basics

- Installing Python, VS Code, and Jupyter
- Understanding the Python interpreter and REPL
- Variables, data types (int, float, str, bool)
- Basic I/O: `print()` and `input()`
- Comments, indentation, and Python style (PEP 8)
- Assignment: Build a "Personal Info" program using variables

## 2

### Week 2 — Control Flow & Logic

- Conditional statements: `if`, `elif`, `else`
- Comparison and logical operators
- `for` loops and `while` loops
- Loop control: `break`, `continue`, `pass`
- Nested conditionals and loops
- Assignment: Build a number-guessing game

## 3

### Week 3 — Data Structures

- Lists: creation, indexing, slicing, methods
- Tuples: immutability and use cases
- Dictionaries: key-value pairs, nested dicts
- Sets: unique values and set operations
- List comprehensions and dictionary comprehensions
- Assignment: Student grade tracker using lists and dicts

## 4

### Week 4 — Functions & Modules

- Defining and calling functions
- Arguments: positional, keyword, default, `*args/**kwargs`
- Return values and scope (local vs. global)
- Lambda functions and higher-order functions
- Importing and using standard library modules
- Mini Project: Build a reusable utility library

# Intermediate Phase: Weeks 5–8

Go beyond syntax. In this phase, you'll write professional-quality Python using object-oriented design, work with real data, connect to APIs, and interact with databases.

1

## Week 5 — OOP: Classes & Objects

- Classes, objects, and instance variables
- The `__init__` method and `self`
- Instance methods vs. class methods vs. static methods
- Inheritance, method overriding, and `super()`
- Encapsulation and abstraction principles
- Assignment: Model a real-world system (e.g., bank account)

2

## Week 6 — File I/O & Error Handling

- Reading and writing text files with `open()`
- Working with CSV files using the `csv` module
- JSON read/write with the `json` module
- Exception handling: `try`, `except`, `finally`
- Custom exceptions and error logging
- Assignment: Build a log file parser

3

## Week 7 — Data Manipulation with Pandas

- DataFrames and Series: creation and structure
- Loading data from CSV, Excel, and JSON
- Filtering, sorting, and grouping data
- Handling missing values and duplicates
- Merging and reshaping datasets
- Assignment: Analyze a real-world dataset (e.g., sales data)

4

## Week 8 — APIs, Web Scraping & Databases

- HTTP basics and the `requests` library
- Consuming REST APIs and parsing JSON responses
- Web scraping with BeautifulSoup
- SQLite3: creating tables and running queries
- Connecting Python to SQL databases with `sqlite3`
- Mini Project: Build a data pipeline from API to database

# Advanced & Application Phase: Weeks 9–12

Apply everything you've learned to solve real business problems. This phase covers automation, data visualization, introductory machine learning, and culminates in a capstone project.

1

## Week 9 — Automation & Scripting

- Automating file system tasks with `os` and `shutil`
- Scheduling scripts with `schedule` library
- Automating Excel and PDF reports with `openpyxl`
- Sending automated emails with `smtplib`
- Browser automation basics with Selenium
- Assignment: Build an automated daily report generator

2

## Week 10 — Data Visualization

- Matplotlib: line, bar, scatter, and histogram plots
- Seaborn: statistical plots and styling themes
- Customizing charts: labels, titles, legends, colors
- Subplots and multi-panel dashboards
- Intro to Plotly for interactive visualizations
- Assignment: Build a visual report from a Pandas dataset

3

## Week 11 — Intro to ML Pipelines

- What is machine learning? Types and terminology
- scikit-learn: loading datasets and preprocessing
- Train/test split and cross-validation
- Building a linear regression model end-to-end
- Evaluating model performance: MSE,  $R^2$ , accuracy
- Assignment: Predict housing prices with a regression model

4

## Week 12 — Capstone Project

- Integrate skills from all three phases
- Choose a domain: finance, healthcare, e-commerce, or ops
- Build a complete data pipeline with visualization and analysis
- Document your project with a README on GitHub
- Present findings in a 15-minute live demo session
- Deliverable: Deployed or shareable portfolio project

## INDUSTRY EXAMPLE 1

# Real-World Use Case: E-Commerce Sales Analytics

## The Business Problem

A mid-size e-commerce company generates thousands of daily transactions. Their team manually exports CSV reports and spends 10+ hours per week building Excel summaries — prone to errors and always delayed.

## Role of Early Professionals

A junior data analyst is tasked with automating the weekly reporting pipeline, reducing manual work and improving turnaround from 48 hours to under 30 minutes.

## How Python Is Applied

- Pandas loads, cleans, and aggregates raw transaction CSVs
- Custom functions calculate KPIs: revenue, AOV, churn rate
- Matplotlib/Seaborn generate weekly trend charts automatically
- openpyxl writes formatted Excel reports with embedded charts
- smtplib delivers the report to stakeholders every Monday at 8 AM

## Measurable Business Impact

- Reporting time reduced by 95% (48 hrs → 30 min)
- Zero manual errors in downstream dashboards
- Analyst freed to focus on strategic analysis and A/B testing

## INDUSTRY EXAMPLE 2

# Real-World Use Case: HR Attrition Prediction

## The Business Problem

A 2,000-person technology firm experiences 18% annual employee turnover — costing an estimated \$8,000–\$15,000 per departing employee in hiring and training. HR leadership wants to predict who is likely to leave before they resign.

## Role of Early Professionals

A people analytics intern builds a supervised ML model using historical HR data and presents findings to the CHRO with a visual dashboard.

## How Python Is Applied

- Pandas processes HR data: tenure, performance scores, satisfaction ratings
- scikit-learn trains a logistic regression and random forest classifier
- Feature importance analysis identifies top attrition drivers
- Seaborn visualizes correlation heatmaps and attrition distributions
- Output: a ranked list of at-risk employees for HR intervention

## Measurable Business Impact

- Model achieves 82% prediction accuracy on holdout data
- HR proactively retained 35 high-risk employees in Q1
- Estimated cost savings: \$280,000+ in the first year

# Real-World Use Case: Financial News Sentiment Monitor

## The Business Problem

A fintech startup needs to monitor real-time market sentiment for 50+ publicly traded companies to support its trading algorithm. Manual news review is not scalable — they need an automated, always-on pipeline.

## Role of Early Professionals

A junior Python developer builds and maintains the data ingestion pipeline, integrating multiple news APIs and storing structured sentiment scores in a database.

## How Python Is Applied

- `requests` fetches articles from NewsAPI and RSS feeds every 15 minutes
- BeautifulSoup parses and cleans raw HTML article content
- TextBlob/VADER performs sentiment scoring on headlines and body text
- SQLite3 stores timestamped sentiment records per ticker symbol
- Plotly generates an interactive live dashboard of sentiment trends

## Measurable Business Impact

- Sentiment signal improves trade timing accuracy by ~12%
- Data latency reduced from 4 hours (manual) to under 15 minutes
- Pipeline processes 500+ articles per day with zero manual intervention

WEEKLY STRUCTURE

# How Each Week Is Structured

Every week follows a consistent rhythm designed to maximize learning while fitting into a working professional's schedule. Sessions are short, focused, and actionable.



 **Estimated Weekly Commitment:** 6-8 hours total, including self-paced practice, live sessions, and assignments. Flexible scheduling available for recorded content.

CAREER ALIGNMENT

# Job Roles Aligned to This Course

Completing this course positions you for a range of high-demand roles across industries. Below are the most direct career paths, along with the key skills each role requires.

## Junior Python Developer

- OOP, functions, modules
- REST API integration
- Git & version control
- Flask or scripting basics

## Data Analyst

- Pandas & NumPy
- Data cleaning & EDA
- Matplotlib & Seaborn
- SQL + Python integration

## Automation Engineer

- File & system scripting
- Scheduling & task automation
- openpyxl, smtplib
- Selenium basics

## ML Engineer (Junior)

- scikit-learn pipelines
- Feature engineering
- Model evaluation metrics
- Jupyter + GitHub workflow

# Skills Map: Beginner → Job Ready

Track your progression across three skill tiers. Each phase of the course directly maps to the next level of professional readiness.

1

## Beginner

Python syntax, variables, loops, functions, basic data structures — the ability to write, run, and debug simple scripts independently.

2

## Intermediate

OOP design, file I/O, API consumption, Pandas data manipulation, SQLite integration — the ability to build multi-file programs and analyze real datasets.

3

## Job Ready

Automation scripts, data visualization, intro ML, GitHub portfolio, capstone demo — the ability to independently deliver Python solutions in a professional team setting.

# Capstone Project: What to Expect

The Week 12 capstone is your opportunity to demonstrate full-stack Python competency. It's designed to be portfolio-worthy and interview-ready.

## Project Requirements

- Solve a real business problem in your chosen domain
- Use at least 3 libraries covered in the course
- Include data ingestion, processing, and output (visual or report)
- Fully documented on GitHub with a professional README
- Presented in a 15-minute live demo with Q&A

## Domain Options

-  Finance: Portfolio tracker or market data analyzer
-  Healthcare: Patient data insights or appointment scheduler
-  E-Commerce: Sales dashboard or recommendation logic
-  Operations: Workflow automation or KPI reporting tool

## Evaluation Criteria

- **Code Quality:** Clean, readable, PEP 8-compliant code
- **Functionality:** Does it run and solve the stated problem?
- **Complexity:** Appropriate use of advanced concepts
- **Presentation:** Clear explanation and live demo
- **Documentation:** README, inline comments, and structure

## Expected Outcomes

A GitHub-hosted project you can share with recruiters, reference in interviews, and continue building upon after the course.

## KEY TAKEAWAYS

# Key Takeaways

Here's what makes this course different — and what you'll walk away with after 12 weeks of focused, practical learning.



## Strong Fundamentals First

70–80% of the course is dedicated to core Python concepts — because real professionals need to write reliable, readable code, not just copy-paste snippets.



## Live Mentoring Every Week

A live 1:1 or small-group session each week ensures no one gets stuck. Doubt clearing and code reviews accelerate your growth significantly.



## Industry Use Cases Throughout

Every phase connects theory to business reality. You'll see exactly how Python is applied in e-commerce, HR analytics, fintech, and more.



## Portfolio You Can Show

You'll finish with a GitHub profile, a capstone project, and weekly assignment outputs — tangible proof of skill for any recruiter or hiring manager.

 **Remember:** The goal isn't just to learn Python — it's to **think like a Python professional**. That means clean code, real problem-solving, and the confidence to contribute from day one on the job.