# **Python Learning Path for Data Engineering (5-Week Program, Weekdays Only)**

| **Days** | **Topic** | **Description** | **Practice Question (Data Engineering Interview Questions)** | **Estimated Time** | **Real-time Usage Scenario** |
| --- | --- | --- | --- | --- | --- |
| **Week 1: Python Fundamentals & Data Structures** |  |  |  |  |  |
| Day 1 | Python Setup & Basics | Installing Python, IDE setup (VSCode/PyCharm), running first program, basic syntax, variables, data types | **Interview Q:** Explain the difference between Python 2 and 3. How would you handle different data types in a data pipeline? | 4 hours | Setting up your development environment for future data projects |
| Day 2 | Control Flow | Conditional statements (if, elif, else), comparison operators, logical operators | **Interview Q:** Write a function that validates data points against business rules. How would you make it configurable? | 4 hours | Building data validation logic for pipelines |
| Day 3 | Loops & Iterations | For loops, while loops, nested loops, break/continue statements, list comprehensions | **Interview Q:** Explain how you would process a large file line by line without loading it entirely into memory. | 4 hours | Processing sequential data records in batch operations |
| Day 4 | Data Structures I | Lists, tuples, accessing elements, slicing, common methods | **Interview Q:** How would you find duplicate records in a large dataset? Explain the time and space complexity. | 4 hours | Storing and manipulating collections of data elements |
| Day 5 | Data Structures II | Dictionaries, sets, their methods and operations | **Interview Q:** Design a data structure to efficiently look up records by multiple attributes. | 4 hours | Creating lookup tables for data references |
| **Week 2: Functions, Error Handling & Files** |  |  |  |  |  |
| Day 6 | Functions I | Defining functions, parameters, return values, scope | **Interview Q:** Write a function to process a batch of data records with validation and transformation steps. | 4 hours | Building reusable code components for data pipelines |
| Day 7 | Functions II | Lambda functions, higher-order functions, decorators | **Interview Q:** How would you implement a retry mechanism for API calls in a data pipeline? | 4 hours | Creating robust data processing components |
| Day 8 | Error Handling | Try-except blocks, raising exceptions, handling different error types | **Interview Q:** Write robust functions that handle potential errors in data processing. How would you log errors? | 4 hours | Creating resilient data pipelines that don't break with bad data |
| Day 9 | File Operations I | Reading/writing text files, context managers, working with paths | **Interview Q:** How would you efficiently process a 10GB text file with limited memory? | 4 hours | Building data ingestion systems for large files |
| Day 10 | File Operations II | CSV handling, JSON processing, working with compressed files | **Interview Q:** Write a function that extracts specific fields from a CSV file while handling potential format issues. | 4 hours | Creating parsers for various data formats |
| **Week 3: OOP & Data Processing Libraries** |  |  |  |  |  |
| Day 11 | OOP Basics | Classes, objects, attributes, methods, constructors | **Interview Q:** Design a class to represent a data record with validation logic. How would you handle different data types? | 4 hours | Building custom data structures for specific business domains |
| Day 12 | OOP Advanced | Inheritance, polymorphism, encapsulation, abstract classes | **Interview Q:** Create a hierarchy of data processor classes for different data sources. Explain your design choices. | 4 hours | Designing extensible data processing frameworks |
| Day 13 | Modules & Packages | Importing modules, creating packages, virtual environments, pip | **Interview Q:** How would you structure a large data engineering project? How would you manage dependencies? | 4 hours | Organizing and reusing code across multiple projects |
| Day 14 | NumPy Basics | Arrays, indexing, broadcasting, mathematical operations | **Interview Q:** Explain how NumPy makes numerical computation more efficient than pure Python. Provide examples. | 4 hours | Performing efficient numerical computations on large datasets |
| Day 15 | Pandas Fundamentals | Series, DataFrames, indexing, filtering, basic operations | **Interview Q:** Describe how you would clean a messy dataset with missing values and inconsistent formatting. | 4 hours | Handling structured data for analysis and transformation |
| **Week 4: Advanced Data Processing & Databases** |  |  |  |  |  |
| Day 16 | Pandas Advanced I | Grouping, aggregation, pivoting, reshaping data | **Interview Q:** Write code to perform a complex aggregation on a large dataset efficiently. | 4 hours | Creating data transformations for business intelligence |
| Day 17 | Pandas Advanced II | Merging, joining, time series data, handling categorical data | **Interview Q:** How would you join two large DataFrames efficiently? What considerations would you make? | 4 hours | Combining and transforming datasets for analysis |
| Day 18 | Data Visualization | Matplotlib, Seaborn basics, creating plots, customizing visualizations | **Interview Q:** How would you visualize data distribution and anomalies to communicate findings to stakeholders? | 4 hours | Creating insightful data visualizations for reporting |
| Day 19 | SQL Basics | SQL syntax, queries, joins, aggregations | **Interview Q:** Write a SQL query to find the top 5 customers by purchase amount with details from multiple tables. | 4 hours | Querying databases for data extraction |
| Day 20 | Database Connectivity | SQLAlchemy, connecting to databases, ORM basics, transactions | **Interview Q:** How would you implement a data pipeline that efficiently extracts data from a database? | 4 hours | Building data extraction processes from databases |
| **Week 5: Data Engineering Essentials & Production** |  |  |  |  |  |
| Day 21 | ETL Concepts | Extract-Transform-Load patterns, data pipeline design | **Interview Q:** Design an ETL pipeline for daily sales data processing. How would you handle late or changed data? | 4 hours | Building core data integration processes |
| Day 22 | Data Quality | Data validation techniques, schema validation, anomaly detection | **Interview Q:** How would you implement data quality checks in a production pipeline? What metrics would you track? | 4 hours | Ensuring data reliability in pipelines |
| Day 23 | Parallel Processing | Multithreading, multiprocessing, concurrent execution | **Interview Q:** Compare different approaches to parallelize a data processing job. When would you use each? | 4 hours | Optimizing resource usage for large-scale data operations |
| Day 24 | Cloud & APIs | Cloud storage, RESTful APIs, working with web services | **Interview Q:** Design a system to collect data from multiple APIs and store it in the cloud. How would you handle rate limits? | 4 hours | Integrating with external data sources and cloud services |
| Day 25 | Capstone Project | End-to-end data engineering project | **Interview Q:** Design and implement a complete data pipeline that ingests, processes, and serves data from multiple sources. | 8 hours | Applying all learned concepts to solve a complex problem |

## **Additional Data Engineering Interview Questions by Topic**

### **Python Fundamentals**

1. What's the difference between a list and a tuple in Python?
2. Explain list comprehensions and provide an example relevant to data processing.
3. How would you optimize memory usage when processing large files in Python?
4. What are generators and how are they useful in data engineering?
5. How do you handle missing or NULL values in datasets?

### **Data Structures & Algorithms**

1. How would you efficiently find common elements between two large datasets?
2. Explain the difference between merge sort and quick sort. When would you use each?
3. How would you implement a cache for frequently accessed data?
4. Describe how you would detect cycles in a directed graph of data dependencies.
5. How would you design a data structure for a real-time analytics dashboard?

### **Databases & SQL**

1. What's the difference between a LEFT JOIN and INNER JOIN? Provide an example.
2. How would you optimize a slow SQL query?
3. Explain database normalization and denormalization. When would you use each?
4. How would you design a database schema for a data warehouse?
5. What are database indexes and how do they work?
6. How would you handle database migrations in a production environment?

### **ETL & Data Pipelines**

1. Describe the ETL process and its components.
2. How would you handle a failed task in the middle of a data pipeline?
3. How would you design an incremental loading process?
4. What strategies would you use to optimize a slow-running ETL job?
5. How would you ensure data consistency across multiple data sources?
6. Explain the difference between batch processing and stream processing.

### **Big Data Technologies**

1. What is Hadoop and how does it work?
2. Explain the difference between map and reduce operations.
3. How does Spark differ from Hadoop? What are its advantages?
4. How would you optimize a Spark job?
5. Explain data partitioning in big data systems.

### **Cloud & DevOps**

1. How would you migrate an on-premises data pipeline to the cloud?
2. What are the trade-offs between different cloud storage options?
3. Explain the concept of Infrastructure as Code and its benefits.
4. How would you implement CI/CD for a data engineering project?
5. How would you monitor a data pipeline in production?

### **System Design**

1. Design a real-time analytics system for e-commerce website data.
2. How would you design a data lake architecture?
3. Design a system to process and analyze log data from multiple sources.
4. How would you design a recommendation system's data pipeline?
5. How would you handle schema evolution in a data warehouse?

### **Problem-Solving Scenarios**

1. Your ETL job is running significantly slower than usual. How would you troubleshoot?
2. You need to join two tables, one very large and one small. What approach would you take?
3. How would you handle a sudden 10x increase in data volume?
4. A critical data pipeline has started producing incorrect results. What steps would you take?
5. How would you implement a data quality monitoring system?

## **Resources for Accelerated Learning**

1. **Essential Books**:  
   * "Python for Data Analysis" by Wes McKinney
   * "Designing Data-Intensive Applications" by Martin Kleppmann
2. **Online Learning**:  
   * DataCamp's Python for Data Engineering (focused tracks)
   * YouTube: Search for "Data Engineering with Python" tutorials
3. **Documentation**:  
   * Official Pandas, NumPy documentation
   * PySpark documentation
   * SQLAlchemy documentation
4. **Interview Preparation**:  
   * LeetCode Database and Python problems
   * DataLemur data engineering interview questions
   * StrataScratch data challenges

## **High-Intensity Learning Tips**

1. **Focused Sessions**: Use the Pomodoro technique - 25 minutes of focused work, 5-minute break.
2. **Apply Immediately**: Implement what you learn the same day, even in simple forms.
3. **Teach Others**: Explain concepts to teammates to solidify understanding.
4. **Build Real Projects**: Try to solve actual business problems with each new skill.
5. **Use Spaced Repetition**: Review core concepts at increasing intervals.
6. **Join a Study Group**: Form a learning circle with teammates for accountability.
7. **Connect Concepts**: Constantly relate new information to previously learned material.
8. **Get Feedback Early**: Have experienced data engineers review your work frequently.