

# 100 Days of Data Science Coding Roadmap

## Foundation Phase: Python Basics (Days 1-15)

### ☐ Day 1: Introduction to Python

- Learn about variables, data types, and basic operations
- Set up Python environment (install Python, choose an IDE)
- Write a program that uses variables of different types and performs basic calculations

### ☐ Day 2: Strings and String Manipulation

- Learn string operations, methods, and formatting
- Write a program that manipulates and formats strings to create a personalized message

### ☐ Day 3: Lists and List Operations

- Learn to create, access, and modify lists
- Write a program that creates a to-do list where items can be added, removed, and displayed

### ☐ Day 4: Dictionaries and Sets

- Learn dictionary creation, access, and methods
- Write a program that stores and retrieves information using dictionaries

### ☐ Day 5: Control Flow - Conditionals

- Learn if, elif, else statements
- Write a program that makes decisions based on user input or data values

### ☐ Day 6: Control Flow - Loops

- Learn for and while loops
- Write a program that processes a list of numbers using loops to calculate statistics

### ☐ Day 7: Functions

- Learn function definition, parameters, return values
- Write a program with multiple functions that solve specific tasks

### ☐ Day 8: Error Handling

- Learn try/except blocks and common exceptions
- Write a program that handles potential errors gracefully

### ☐ Day 9: File Operations

- Learn reading from and writing to files
- Write a program that reads data from a file, processes it, and writes results to another file

### ☐ Day 10: Modules and Libraries

- Learn importing modules and using built-in libraries
- Write a program that uses Python's standard libraries to solve a problem

### ☐ **Day 11: List Comprehensions**

- Learn efficient list creation with comprehensions
- Write a program that uses list comprehensions to transform data

### ☐ **Day 12: Lambda Functions**

- Learn anonymous functions and their uses
- Write a program that uses lambda functions for data filtering and transformation

### ☐ **Day 13: Object-Oriented Programming I**

- Learn classes, objects, attributes, and methods
- Write a program that defines a class and creates instances to model real-world entities

### ☐ **Day 14: Object-Oriented Programming II**

- Learn inheritance and polymorphism
- Write a program that uses inheritance to create specialized classes

### ☐ **Day 15: Python Mini-Project**

- Apply all basic Python concepts in a comprehensive project
- Create a command-line application that manages and analyzes a dataset

## **Data Analysis Phase (Days 16-40)**

### ☐ **Day 16: Introduction to NumPy**

- Learn numpy arrays and basic operations
- Write a program that performs calculations on arrays of numbers

### ☐ **Day 17: NumPy Array Operations**

- Learn advanced array manipulation and mathematical functions
- Write a program that uses NumPy to solve a system of equations

### ☐ **Day 18: Introduction to Pandas**

- Learn Series and DataFrame objects
- Write a program that creates and manipulates a simple DataFrame

### ☐ **Day 19: Pandas Data Manipulation**

- Learn filtering, selecting, and transforming data
- Write a program that cleans and prepares a messy dataset

### ☐ **Day 20: Data Cleaning with Pandas**

- Learn handling missing data and data validation
- Write a program that identifies and fixes data quality issues

### ☐ **Day 21: Data Analysis with Pandas**

- Learn groupby operations and aggregations
- Write a program that performs group-wise calculations on a dataset

## ☐ **Day 22: Time Series Analysis Basics**

- Learn date/time handling in Pandas
- Write a program that processes and analyzes time-stamped data

## ☐ **Day 23: Introduction to Data Visualization**

- Learn basic plotting with Matplotlib
- Write a program that creates simple plots from a dataset

## ☐ **Day 24: Advanced Matplotlib**

- Learn customizing plots and multiple subplots
- Write a program that creates publication-quality visualizations

## ☐ **Day 25: Introduction to Seaborn**

- Learn statistical visualizations
- Write a program that uses Seaborn to visualize distributions and relationships

## ☐ **Day 26: Interactive Visualizations with Plotly**

- Learn creating interactive charts
- Write a program that builds interactive visualizations for a dataset

## ☐ **Day 27: Data Aggregation and Pivoting**

- Learn pivot tables and cross-tabulations
- Write a program that reshapes data for analysis and visualization

## ☐ **Day 28: Merging and Joining Datasets**

- Learn combining multiple data sources
- Write a program that performs a data analysis requiring multiple datasets

## ☐ **Day 29: Statistical Functions in Python**

- Learn descriptive statistics and hypothesis testing
- Write a program that calculates statistics and interprets results

## ☐ **Day 30: Exploratory Data Analysis (EDA)**

- Learn systematic approaches to exploring datasets
- Write a program that performs a complete EDA on a dataset

## ☐ **Day 31: Introduction to SQL**

- Learn SQL queries with SQLite in Python
- Write a program that creates, queries, and updates a database

## ☐ **Day 32: Advanced SQL and Database Operations**

- Learn joins, subqueries, and transactions
- Write a program that performs complex database operations

## ☐ **Day 33: Web Scraping Basics**

- Learn HTML parsing with BeautifulSoup

- Write a program that extracts information from a web page

#### ☐ **Day 34: API Interactions**

- Learn making HTTP requests and parsing responses
- Write a program that fetches and processes data from a public API

#### ☐ **Day 35: Working with JSON and CSV**

- Learn parsing and generating structured data files
- Write a program that converts between different data formats

#### ☐ **Day 36: Regular Expressions**

- Learn pattern matching in text
- Write a program that extracts information using regex

#### ☐ **Day 37: Introduction to Data Cleaning and Preparation**

- Learn comprehensive data preprocessing workflows
- Write a program that prepares a raw dataset for analysis

#### ☐ **Day 38: Feature Engineering Basics**

- Learn creating new features from existing data
- Write a program that transforms a dataset to improve its analytical value

#### ☐ **Day 39: Data Analysis Mini-Project I**

- Apply data analysis concepts to a real-world dataset
- Perform EDA, create visualizations, and document insights

#### ☐ **Day 40: Data Analysis Mini-Project II**

- Create a comprehensive data analysis report
- Present findings and recommendations based on data

## **Machine Learning Phase (Days 41-75)**

#### ☐ **Day 41: Introduction to Machine Learning**

- Learn fundamental concepts and terminology
- Write a program that loads and explores a standard ML dataset

#### ☐ **Day 42: Introduction to Scikit-learn**

- Learn the Scikit-learn API and common patterns
- Write a program that uses Scikit-learn to process a dataset

#### ☐ **Day 43: Linear Regression**

- Learn simple and multiple linear regression
- Write a program that builds and evaluates a linear regression model

#### ☐ **Day 44: Model Evaluation for Regression**

- Learn regression metrics and validation techniques

- Write a program that assesses regression model performance

#### ☐ **Day 45: Logistic Regression**

- Learn binary classification fundamentals
- Write a program that builds and evaluates a logistic regression model

#### ☐ **Day 46: Model Evaluation for Classification**

- Learn classification metrics and confusion matrices
- Write a program that assesses classification model performance

#### ☐ **Day 47: Decision Trees**

- Learn tree-based models and their interpretation
- Write a program that builds and visualizes a decision tree

#### ☐ **Day 48: Random Forests**

- Learn ensemble methods and their advantages
- Write a program that compares decision trees to random forests

#### ☐ **Day 49: Support Vector Machines**

- Learn SVM theory and applications
- Write a program that implements SVM for classification

#### ☐ **Day 50: Naive Bayes**

- Learn probabilistic classification
- Write a program that implements Naive Bayes for text classification

#### ☐ **Day 51: K-Nearest Neighbors**

- Learn distance-based learning
- Write a program that implements KNN for classification and regression

#### ☐ **Day 52: Cross-Validation**

- Learn robust model evaluation techniques
- Write a program that uses cross-validation to tune model parameters

#### ☐ **Day 53: Hyperparameter Tuning**

- Learn GridSearch and RandomizedSearch
- Write a program that finds optimal parameters for a model

#### ☐ **Day 54: Feature Selection**

- Learn methods for choosing relevant features
- Write a program that selects the best features for a model

#### ☐ **Day 55: Dimensionality Reduction with PCA**

- Learn Principal Component Analysis
- Write a program that applies PCA to visualize high-dimensional data

#### ☐ **Day 56: Clustering with K-means**

- Learn unsupervised learning basics
- Write a program that identifies clusters in a dataset

#### ☐ **Day 57: Hierarchical Clustering**

- Learn top-down and bottom-up clustering
- Write a program that builds and visualizes a dendrogram

#### ☐ **Day 58: DBSCAN Clustering**

- Learn density-based clustering
- Write a program that applies DBSCAN to detect clusters of arbitrary shape

#### ☐ **Day 59: Anomaly Detection**

- Learn techniques for identifying outliers
- Write a program that detects anomalies in a dataset

#### ☐ **Day 60: Model Interpretation Basics**

- Learn feature importance and model explanation
- Write a program that explains a model's predictions

#### ☐ **Day 61: Introduction to Neural Networks**

- Learn perceptrons and multi-layer networks
- Write a program that builds a simple neural network

#### ☐ **Day 62: Introduction to TensorFlow or PyTorch**

- Learn basics of a deep learning framework
- Write a program that implements a neural network for classification

#### ☐ **Day 63: Deep Learning for Classification**

- Learn neural network architectures for classification
- Write a program that builds and trains a classification network

#### ☐ **Day 64: Deep Learning for Regression**

- Learn neural network architectures for regression
- Write a program that builds and trains a regression network

#### ☐ **Day 65: Convolutional Neural Networks Basics**

- Learn CNN architecture and applications
- Write a program that builds a simple CNN for image classification

#### ☐ **Day 66: Transfer Learning**

- Learn using pre-trained models
- Write a program that applies transfer learning to a new problem

#### ☐ **Day 67: Recurrent Neural Networks Basics**

- Learn sequence modeling with RNNs
- Write a program that builds a simple RNN for sequence prediction

#### ☐ **Day 68: Natural Language Processing Basics**

- Learn text preprocessing techniques
- Write a program that processes and analyzes text data

#### ☐ **Day 69: Sentiment Analysis**

- Learn text classification for sentiment
- Write a program that determines sentiment in text samples

#### ☐ **Day 70: Word Embeddings**

- Learn word vector representations
- Write a program that uses pre-trained word embeddings

#### ☐ **Day 71: Time Series Forecasting**

- Learn methods for predicting future values
- Write a program that builds a time series forecast model

#### ☐ **Day 72: Recommendation Systems**

- Learn collaborative and content-based filtering
- Write a program that builds a simple recommendation system

#### ☐ **Day 73: Introduction to Model Deployment**

- Learn saving/loading models and creating APIs
- Write a program that deploys a trained model as a web service

#### ☐ **Day 74: Machine Learning Mini-Project I**

- Apply ML concepts to a real-world problem
- Build and evaluate multiple models for comparison

#### ☐ **Day 75: Machine Learning Mini-Project II**

- Complete a comprehensive ML solution
- Document model development, evaluation, and findings

### **Advanced Topics and Capstone Projects (Days 76-100)**

#### ☐ **Day 76: Introduction to Big Data**

- Learn distributed computing concepts
- Write a program that processes a large dataset efficiently

#### ☐ **Day 77: Spark Basics**

- Learn PySpark fundamentals
- Write a program that performs distributed data processing

#### ☐ **Day 78: Spark for Machine Learning**

- Learn ML with PySpark
- Write a program that builds a distributed ML model

### ☐ **Day 79: Introduction to MLOps**

- Learn ML operations principles
- Write a program that implements a basic ML pipeline

### ☐ **Day 80: Model Monitoring and Maintenance**

- Learn techniques for model deployment lifecycle
- Write a program that monitors model performance over time

### ☐ **Day 81: A/B Testing**

- Learn statistical experiment design
- Write a program that analyzes A/B test results

### ☐ **Day 82: Bayesian Statistics and PyMC**

- Learn Bayesian thinking and MCMC
- Write a program that performs Bayesian inference

### ☐ **Day 83: Reinforcement Learning Basics**

- Learn RL concepts and terminology
- Write a program that implements a simple RL agent

### ☐ **Day 84: Explainable AI (XAI)**

- Learn tools for understanding model decisions
- Write a program that explains complex model predictions

### ☐ **Day 85: Ethics in Data Science**

- Learn fairness, accountability, and transparency
- Write a program that audits a model for bias

### ☐ **Day 86: Cloud Services for Data Science**

- Learn using AWS, GCP, or Azure for data science
- Write a program that leverages cloud services for a data task

### ☐ **Day 87: Docker for Data Science**

- Learn containerization basics
- Write a program that runs in a Docker container

### ☐ **Day 88: Data Science Project Management**

- Learn project organization and documentation
- Write a program that follows best practices for reproducibility

### ☐ **Day 89: Dashboards with Streamlit or Dash**

- Learn interactive dashboard creation
- Write a program that creates a web dashboard for data visualization

### ☐ **Day 90: Automated ML (AutoML)**

- Learn techniques for automating ML workflows



- Write a program that uses AutoML to build an optimal model
- ☐ **Day 91: Capstone Project - Planning**
  - Define a comprehensive data science project
  - Write a detailed project plan and requirements
- ☐ **Day 92: Capstone Project - Data Collection**
  - Gather and organize data for your project
  - Write code to collect, clean, and prepare data
- ☐ **Day 93: Capstone Project - Exploratory Analysis**
  - Analyze and visualize your project data
  - Write code to gain insights from your dataset
- ☐ **Day 94: Capstone Project - Feature Engineering**
  - Prepare data for modeling
  - Write code to create and select optimal features
- ☐ **Day 95: Capstone Project - Modeling I**
  - Develop initial models
  - Write code to train and evaluate multiple approaches
- ☐ **Day 96: Capstone Project - Modeling II**
  - Refine and optimize models
  - Write code to tune and enhance model performance
- ☐ **Day 97: Capstone Project - Model Deployment**
  - Make your model accessible
  - Write code to deploy your solution as a usable product
- ☐ **Day 98: Capstone Project - Documentation**
  - Document your entire project
  - Write comprehensive documentation including results and limitations
- ☐ **Day 99: Capstone Project - Presentation**
  - Prepare to present your work
  - Create visualizations and explanations of your project journey
- ☐ **Day 100: Data Science Journey Reflection**
  - Review your progress and learning
  - Write a reflection on your 100-day journey and plan next steps

## Resources for Your Journey

### Recommended Books:

- "Python for Data Analysis" by Wes McKinney

- "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron
- "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
- "Data Science from Scratch" by Joel Grus

### **Online Resources:**

- Kaggle: For datasets and competitions
- GitHub: For code examples and projects
- Stack Overflow: For troubleshooting
- Documentation for Python libraries (NumPy, Pandas, Scikit-learn, etc.)
- Medium/Towards Data Science: For articles and tutorials

### **Practice Platforms:**

- Kaggle Competitions
- LeetCode (for Python practice)
- DataCamp
- HackerRank

Good luck on your data science journey! Remember to practice consistently and build projects to reinforce your learning.