


# Python Engineer Roadmap

 README to PDF passing

[Python](#) can be used in a lot of computer science fields. In this repository, we have collected resources for each field of computer science that are related to Python.

**Not sure which source to choose?** You can follow the resources marked with a  symbol, they are highly recommended by the community.

## Contribution

Before you head over, read the [Contribution Guide](#) first. You are new to contribution process? For more information about the steps and guides, check out the [First Contribution Guide](#).

## Table of Contents

- [Prerequisites](#)
  - [Algorithms and Data Structures](#)
  - [System Design](#)
  - [Git](#)
  - [Operating System](#)
  - [Virtual Environment](#)
  - [Python](#)
- [Career Path](#)
  - [Backend](#)
    - [Django](#)
    - [FastAPI](#)
    - [Flask](#)
    - [Tornado](#)
    - [Web2Py](#)
    - [Sanic](#)
    - [AIOHTTP](#)
    - [Bottle](#)
    - [Dash](#)
  - [Data Science](#)
    - [Data Analysis](#)
      - [Numpy](#)
      - [Scipy](#)
      - [Pandas](#)
    - [Data Visualization](#)
      - [Matplotlib](#)

- Plotly
- Machine Learning
- Deep Learning
- Neural Networks
- Image Processing
- DevOps
- Hacking
- Algorithmic Trading
- Bot
  - Web
  - Telegram
    - Telethon
    - Pyrogram
    - Python Telegram Bot
    - AIOGram
    - PyTelegramBotApi
    - TeleBot
  - Discord
- Advanced Topics
  - Databases
    - General
    - SQLite
    - PostgreSQL
    - MySQL
    - Oracle
    - MongoDB
    - Redis
    - MemCached
    - Apache Cassandra
  - Clean Code
  - Clean Architecture
  - Caching
  - Testing
    - Python `unittest` Package
    - `PyTest` Testing Package
    - DRF Test Framework
  - Container Platforms
    - Docker
    - Kubernetes
  - Programming Paradigms
    - Object-Oriented Programming

- [Functional Programming](#)
- [Architectural Patterns](#)
  - [Microservice](#)
  - [Enterprise Applications](#)
- [Design Principles](#)
  - [SOLID](#)
  - [KISS](#)
  - [DRY](#)
- [Design Patterns](#)
- [Message Brokers](#)
  - [RabbitMQ](#)
- [WSGI Servers](#)
  - [Gunicorn](#)
  - [uWSGI](#)
- [ASGI Servers](#)
  - [Uvicorn](#)
  - [Starlette](#)
- [Web Servers](#)
  - [Nginx](#)
  - [Apache](#)
- [Availability and Reliability](#)
- [Distributed Systems](#)
- [Reactive Systems](#)
- [Refactoring](#)
- [Security](#)
- [Monitoring](#)
- [Soft Skill](#)
- [Public Cloud](#)
- [Where to Go Next?](#)

## Prerequisites

---

- **Algorithms and Data Structures**

- **Book**

- [✓ Grokking Algorithms by Aditya Bhargava](#)
    - [✓ Cracking the Coding Interview by Gayle Laakmann McDowell](#)
    - [Data Structure and Algorithmic Thinking with Python: Data Structure and Algorithmic Puzzles by Narasimha Karumanchi](#)
    - [Introduction to Algorithms \(CLRS\)](#)

- **Video**

- [Data Structures And Algorithms Course by Mosh Hamedani](#)

- **Platform**

- [✔ LeetCode](#)
- [HackerRank](#)
- [CodeWars](#)
- [TheAlgorithms](#)
- [Codeforces](#)

- **Repo**

- [✔ Coding University](#)

- **Cheat Sheet**

- [Data Structures Cheat Sheet](#)

- **System Design**

- **Book**

- [✔ System Design Interview by Alex Xu](#)
- [System Design Interview: Volume 2 by Alex Xu & Sahn Lam](#)

- **Course**

- [Educative - Grokking the System Design Interview](#)

- **Git**

- **Documentation**

- [Git Documentation](#)

- **Video**

- [✔ Git Course by Mosh Hamedani](#)
- [Git and GitHub tutorial for beginners by Amigoscode](#)
- [Git and GitHub crash course by freeCodeCamp](#)
- [Git For Professionals course by Trevor Miller](#)

- **Book**

- [Pro Git](#)
- [Git Notes for Professionals](#)

- **Website**

- [Git For Beginners by tutorialspoint](#)
- [Git For Intermediates and Professionals by W3schools](#)

- [Advanced Git Tips by atlassian](#)
- [Learn Git Branching](#)

- **Cheat Sheet**

- [Git cheat sheet](#)

- **Operating System**

- **Book**

- [✓ LPIC-1: Linux Professional Institute Certification Study Guide: Exams 101 and 102 by Roderick W. Smith](#)

- **Video**

- [Lpic-1 Course by Jadi](#)

- **Platform**

- [Linux Journey](#)

- **Community**

- [Discord Linux group](#)
- [Linux For Everyone Community](#)

- **Cheat Sheet**

- [Linux commands cheat sheet](#)

- **Virtual Environment**

- **VENV**

- **Documentation**
  - [VENV Documentation](#)

- **VirtualEnvWrapper**

- **Documentation**
  - [VirtualEnvWrapper Documentation](#)

- **PipEnv**

- **Documentation**
  - [PipEnv Documentation](#)

- **Conda**

- **Documentation**
  - [Conda Documentation](#)

- **Python-Poetry**
  - **Documentation**
    - [Python-Poetry Documentation](#)

- **Python**

- **Documentation**
  - [Python Documentation](#)
- **Beginner**
  - **Book**
    - ✓ [Python Crash Course by Eric Matthes](#)
    - [Head First Python by Paul Barry](#)
    - [Learn Python the Hard Way by Zed Shaw](#)
    - [Essential Python Tools](#)
  - **Video**
    - [Python Beginner Tutorial by NeuralNine](#)
    - [Python Programming Tutorials by Tech with Tim](#)
  - **Platform**
    - [W3schools](#)
    - [Codecademy Python 2](#)
    - [Codecademy Python 3](#)
    - [Sololearn Python](#)
- **Intermediate**
  - **Book**
    - ✓ [Python Cookbook by David Beazley & Brian Jones](#)
    - [Beyond the Basic Stuff with Python Best Practices for Writing Clean Code by Sweigart, Al](#)
    - [Fluent Python by Luciano Ramalho 2nd Edition](#)
    - [Effective Python by Brett Slatkin](#)
    - [Python Concurrency with asyncio](#)
  - **Video**
    - [Python Intermediate Tutorial by NeuralNine](#)
    - [Intermediate Python Tutorials by Tech with Tim](#)
  - **Platform**
    - [GeeksForGeeks](#)

- [Programiz](#)

- **Advanced**

- **Book**

- ✓ [Architecture Patterns with Python](#) by Harry Percival & Bob Gregory
    - ✓ [Practices of the Python Pro](#) by Dane Hillard
    - ✓ [Python Tricks](#) by Dan Bader
    - [Python Testing with pytest](#) by Brian Okken
    - [Python Concurrency with asyncio](#) by Matthew Fowler
    - [Python for Programmers](#) by Deitel Developer Series
    - [Serious Python](#) by Julien Danjou
    - [Python Notes for Professionals](#)

- **Video**

- [Python3: Deep Dive \(4 Parts\)](#)

- **Platform**

- ✓ [RealPython](#)
    - [Python-Course](#)

- **Community**

- [Python Discord group](#)
  - [Python Telegram group](#)

- **Cheat Sheet**

- [Beginners Python cheat sheet](#)
  - [Python cheat sheet](#)

## Career Path

---

- **Backend**

- **Django**

- **Documentation**

- [Django Documentation](#)

- **Book**

- [Django for Beginners](#) by William S. Vincent
    - [Django for APIs](#) by William S. Vincent
    - [Django for Professionals](#) by William S. Vincent

- [Two Scoops of Django 3.x](#) by Daniel Roy Greenfeld, Audrey Roy Greenfeld
- [Test-Driven Development with Python: Obey the Testing Goat: Using Django, Selenium, and JavaScript](#) by Harry Percival
- [Test-Driven Development with Django](#) by Kevin Harvey
- [Django3 by example](#) by antonio mele

#### ▪ **Video**

- [Django Web Framework - Full Course for Beginners](#) by Justin Mitchel
- [Build REST APIs with Django REST Framework and Python](#) By Shubham Sarda
- [Django For Everybody - Full Course](#) by Dr. Charles Severance
- [Django ORM Mastery](#) - Very Academy
- [Learn Django Class Base View](#) - Very Academy
- [Django Course](#) by Mosh Hamedani
- [Try Django 3.2 - Python Web Development Tutorial Series](#) by Justin Mitchel

#### ▪ **Awesome Django**

- [Awesome Django](#)

#### ▪ **Community**

- [Django Discord group](#)
- [Django Telegram group](#)

#### ▪ **Cheat Sheet**

- [Django Cheat Sheet](#)
- [Django Models Cheat Sheet](#)
- [Django Class Based Views Cheat Sheet](#)

### ◦ **FastAPI**

#### ▪ **Video**

- [Python API Development - Comprehensive Course for Beginners](#) by Sanjeev Thiagarajan
- [FastAPI course](#) by testdriven.io & talkpython.fm

#### ▪ **Documentation**

- [FastAPI documentation](#)
- [FastAPI Utilities documentation](#)

#### ▪ **Awesome FastAPI**

- [Awesome FastAPI](#)

#### ▪ **Community**

- [FastApi Discord group](#)



- **Flask**

- **Book**

- [Flask Web Development: Developing Web Applications with Python](#)
    - [Flask Framework Cookbook](#)

- **Video**

- [Flask Tutorial by Tech With Tim](#)
    - [REST APIs with Flask and Python by Jose Salvatierra](#)

- **Documentation**

- [Flask Document](#)

- **Cheat Sheet**

- [Flask Cheat Sheet and Quick Reference](#)

- **Tornado**

- **Book**

- [Introduction to Tornado by Michael Dory](#)

- **Video**

- [Tornado, Coroutines and Concurrency by Bek Brace](#)
    - [Tornado in Depth by Oscar Vilaplana](#)
    - [More than just a pretty web framework, the Tornado IOLoop by Gavin M.Roy](#)

- **Documentation**

- [Tornado Document](#)

- **Web2Py**

- **Documentation**

- [Web2Py Document](#)

- **Book**

- [Complete Reference Manual by Massimo Di Pierro](#)
    - [Killer Web Development by Marco Laspe](#)

- **Sanic**

- **Documentation**

- [Sanic Document](#)
    - [Sanic Guide](#)

- **Article**

- [Getting Started with Sanic for Python by Mukul Khanna](#)

- **Video**

- [Async Web Apps with Sanic by Dougal Matthews](#)

- **AIOHTTP**

- **Documentation**

- [AIOHTTP Document](#)

- **Bottle**

- **Documentation**

- [Bottle Document](#)

- **Dash**

- **Documentation**

- [Dash-Python Document](#)

- **Awesome Dash**

- [Awesome Dash](#)

- **Template**

- [Dash Clean Architecture Template](#)

- [List Of All Python Backend Web Frameworks](#)

- **Data Science**

- **Data Analysis**

- **Numpy**

- **Documentation**

- [Numpy Document](#)

- **Scipy**

- **Documentation**

- [Scipy Document](#)

- **Pandas**

- **Documentation**

- [Pandas Document](#)

- **Data Visualization**

- **Matplotlib**

- **Documentation**

- [Matplotlib Document](#)

- **Plotly**

- **Documentation**

- [Plotly Document](#)

- **Machine Learning**

- **Video**

- [Machine Learning Course by Andrew Ng](#)

- **Deep Learning**

- **Neural Networks**

- **Image Processing**

- **DevOps**

- **Community**

- [DevOps, SRE, & Infrastructure](#)

- **Hacking**

- **Book**

- [Black Hat Python, 2nd Edition: Python Programming for Hackers and Pentesters](#)

- **Algorithmic Trading**

- **Bot**

- **Web**

- **Telegram**

- **Telethon**

- [Source](#)

- [Documentation](#)

- **Pyrogram**

- [Source](#)
- [Documentation](#)
- **Python Telegram Bot**
  - [Source](#)
  - [Documentation](#)
- **AIOGram**
  - [Source](#)
  - [Documentation](#)
- **PyTelegramBotApi**
  - [Source](#)
  - [Documentation](#)
- **TeleBot**
  - [Source](#)

- **Discord**

## Advanced Topics

---

⚠ The following topics don't have any order or priority of learning.

☐ Choose topics that you are **interested in** or **suit your needs**.

- **Databases**

- **General**

- **Book**

- [✓ Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems](#)
- [Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement](#)

- **SQLite**

- **Documentation**

- [SQLite Documentation](#)

- **Tutorials**

- [SQLite Tutorial](#)

- **PostgreSQL**

- **Documentation**

- [PostgreSQL Documentation](#)

- **Community**

- [PostgreSQL Telegram group](#)

- **Cheat Sheet**

- [PostgreSQL Cheat Sheet By GoalKicker](#)
    - [PostgreSQL CHEAT SHEET](#)
    - [POSTGRESQL 8.3 PSQL CHEAT SHEET](#)

- **MySQL**

- **Documentation**

- [MySQL Documentation](#)

- **Tutorials**

- [MySQL Tutorial](#)

- **Cheat Sheet**

- [MySQL Cheat Sheet by GoalKicker](#)

- **Oracle**

- **Documentation**

- [Oracle Documentation](#)

- **Tutorials**

- [Oracle Tutorial](#)

- **Cheat Sheet**

- [Oracle Cheat Sheet by GoalKicker](#)

- **MongoDB**

- **Documentation**

- [MongoDB Documentation](#)

- **Tutorial**

- [Python MongoDB](#)

- **Cheat Sheet**

- [MongoDB Cheat Sheet by GoalKicker](#)

- **Redis**

- **Documentation**

- [Redis Documentation](#)

- **Article**

- [How to use redis with python by Brad Solomon](#)

- **Cheat Sheet**

- [Redis Cheat Sheet 1](#)
    - [Redis Cheat Sheet 2](#)

- **MemCached**

- **Documentation**

- [MemCached Wiki](#)

- **Apache Cassandra**

- **Documentation**

- [Apache Cassandra Documentation](#)

- **Clean Code**

- **Book**

- ✓ [Clean Code in Python by Mariano Anaya](#)
    - [Code Complete: A Practical Handbook of Software Construction, Second Edition by Steve McConnell](#)

- **Clean Architecture**

(In Progress)

- **Caching**

- **Article**

- [Caching in django with redis by Real Python](#)

- **Video**

- [Redis Course - In-Memory Database Tutorial](#)

- **Testing**

- **Python `unittest` package**
  - **Documentation**
    - [Testing in Python](#)
    - [Getting Started With Testing in Python](#)
- **PyTest Testing Package**
  - **Documentation**
    - [Official PyTest Documentation](#)
- **DRF Test Framework**
  - **Documentation**
    - [Testing - Django REST framework](#)
  - **Video**
    - [Pytest Django and Django Rest Framework](#)

## • **Container Platforms**

- **Docker**
  - **Documentation**
    - [Docker Documentation](#)
  - **Book**
    - [Docker in Action, Second Edition](#)
    - [Docker Deep Dive: Zero to Docker in a single book](#)
  - **Video**
    - [Docker Mastery With Django - very academy](#)
    - [Docker Course by Mosh Hamedani](#)
    - [Docker Swarm Step by Step](#)
  - **Cheat Sheet**
    - [Docker Cheat Sheet](#)
- **Kubernetes**
  - **Documentation**
    - [Kubernetes Documentation](#)
  - **Video**

- ["Just me and Opensource" YouTube channel](#)
- **Book**
  - [Kubernetes: Up and Running, 2nd Edition](#)
  - [Kubernetes in Action, Second Edition](#)
- **Community**
  - [Kubernetes Discord group](#)

## • Programming Paradigms

- **Object-Oriented Programming**
  - **Book**
    - [Python3 Object-Oriented Programming](#)
- **Functional Programming**
  - **Article**
    - [Functional Programming in Python](#)

## • Architectural Patterns

- **Microservice**
  - **Book**
    - [Microservice Architecture](#)
    - [Building Microservices, 2nd Edition](#)
- **Enterprise Applications**
  - **Book**
    - [Patterns of Enterprise Application Architecture](#)
    - [Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions](#)

## • Design Principles

- **SOLID**
  - **Article**
    - [SOLID Principles In Python by Philip Norton](#)
    - [Uncle Bob's SOLID principles made easy ☐ - in Python!](#)
- **KISS**
- **DRY**



## • Design Patterns

### ◦ Book

- [Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software](#)
- [Django Design Patterns and Best Practices \(by Arun Ravindran\)](#)
- [Dive Into Design Patterns by Alexander Shvets](#)

### ◦ Video

- [Design Patterns Course by Mosh Hamedani](#)

## • Message Brokers

### ◦ RabbitMQ

#### ▪ Article

- [RabbitMQ Hello World](#)
- [RabbitMQ Work Queues](#)
- [RabbitMQ Publish/Subscribe](#)
- [RabbitMQ Routing](#)
- [RabbitMQ Topics](#)
- [RabbitMQ Remote procedure call \(RPC\)](#)

## • WSGI Servers

### ◦ Gunicorn

#### ▪ Documentation

- [Gunicorn Documentation](#)

### ◦ uWSGI

#### ▪ Documentation

- [uWSGI Documentation](#)

## • ASGI Servers

### ◦ Uvicorn

#### ▪ Documentation

- [Uvicorn Documentation](#)

### ◦ Starlette

#### ▪ Documentation

- [Starlette Documentation](#)

- **Web Servers**

- **Nginx**

- **Documentation**

- [NGINX Documentation](#)

- **Book**

- [NGINX Cookbook](#)

- **Cheat Sheet**

- [NGINX CHEAT SHEET](#)

- **Apache**

- **Documentation**

- [Apache Documentation](#)

- **Book**

- [Apache Cookbook: Solutions and Examples for Apache Administrators](#)

- **Cheat Sheet**

- [Apache Cheat Sheet](#)

- **Availability and Reliability**

(In Progress)

- **Distributed Systems**

(In Progress)

- **Reactive Systems**

(In Progress)

- **Refactoring**

(In Progress)

- **Security**

- **Book**

- [The Web Application Hacker's Handbook](#)

- [Application\\_Security\\_Program\\_Handbook](#) - a guide for software engineers

- [Designing secure software - a guide for developers](#)
- [Hacking APIs Breaking Web Application Programming Interfaces](#)
- [Web Application Security Exploitation and Countermeasures for Modern Web Applications](#)
- [Web Security for Developers: Real Threats, Practical Defense](#)
- [Mastering Modern Web Penetration Testing](#)

- **WebSite**

- [OWASP Top 10](#)
- [OWASP Top 10 for Web with live training](#)
- [SANS SWAT Checklist](#)

- **Monitoring**

(In Progress)

- **Soft Skill**

(In Progress)

- **Public Cloud**

(In Progress)

## Where to Go Next?

(In Progress)

