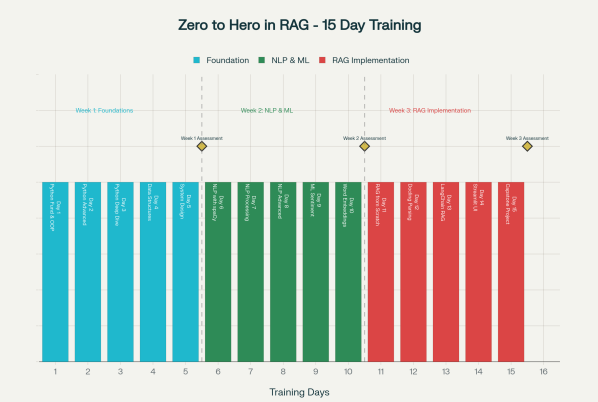
Zero to Hero in Retrieval Augmented Generation
15-Day Intensive Training Program

Program at a Glance



From Zero to Hero in RAG

Learning Path: Python → DSA → System Design → NLP → RAG → Prod

Program Overview

- Duration: 15 working days (3 weeks)
- Daily: 8 hours (4 learning + 4 coding)
- Target: Software engineers with basic programming knowledge

What You'll Build

- 15 progressive projects
- 1 production capstone
- Complete end-to-end RAG

Portfolio Highlights:

- Week 1: 5 Python projects + System design
- Week 2: NLP pipeline + ML model + Embeddings
- Week 3: Production RAG system with UI

Program Architecture

Phase 1: Foundations (Days 1-5)

- Python Fundas
- Advanced Python
- DSA Algorithms
- System Design

Phase 2: NLP & ML (Days 6-10)

- spaCy NLP
- Text Processing
- Sentiment Analysis
- Word Embeddings

Phase 3: RAG (Days 11-15)

- RAG from Scratch
- Docling Parsing
- LangChain RAG
- Streamlit UI

Daily Structure:

- 09:00-13:00: Learning (concept + guided examples)
- 14:00-18:00: Coding (project implementation)
- Thursday 16:00-17:00: Weekly QnA Session

Day 1: Python Fundamentals & OOP

Morning Session (4 hrs)

- Variables & data types
- Control flow (if, loops)
- Functions & scope
- Object-oriented programming
- Inheritance & polymorphism

Afternoon: Pick 1 Project

- Shopping Cart System
- Employment Hierarchy
- Library Management
- Game tic-tac-toe
- Banking System

Resources (Free):

- Google's Python Class (https://developers.google.com/edu/python)
- Python Essentials (https://pythoninstitute.org/)
- Real Python OOP (realpython.com)

Difficulty: Medium |**Time:** 4 hours |**Deliverable:** Working code

Day 2: Advanced Python & File Handling

Morning Session (4 hrs)

- Decorators
- Generators & iterators
- Context managers
- Exception handling
- File I/O, JSON, CSV

Afternoon: Pick 1 Project

- Logging decorators
- CSV data processor
- File backup system
- Config file manager
- Validation decorators

Resources (Free):

- Real Python Decorators (realpython.com/decorators)
- W3Schools File Handling (w3schools.com/python/file)
- DataCamp Data Processing (freemium)

Difficulty: Medium |**Time:** 4 hours |**Deliverable:** Working code

Day 3: Python Advanced Deep Dive

Morning Session (4 hrs)

- Metaclasses
- Descriptors
- Comprehensions
- Async/await
- Memory management
- Performance optimization

Afternoon: Pick 1 Project

- Async web scraper
- Lazy-loading properties
- Concurrent file processor
- Generator ETL pipeline
- Thread-safe cache

Resources (Free):

- Real Python Concurrency (realpython.com/concurrency)
- Fluent Python concepts (github.com/fluentpython)
- Memory Management (realpython.com/memory)

Difficulty: Medium |**Time:** 4 hours |**Deliverable:** Working code

Day 4: Data Structures & Algorithms

Morning Session (4 hrs)

- Arrays, linked lists
- Stacks, queues
- Sorting algorithms
- Searching algorithms
- Big O complexity
- Time/space analysis

Afternoon: Pick 1 Project

- Linked list (LeetCode 206)
- 5 LeetCode problems
- Sorting algorithms
- Expression evaluation
- Hash map problems

Resources (Free):

- W3Schools DSA (w3schools.com/dsa)
- GeeksforGeeks (geeksforgeeks.org/dsa)
- LeetCode Explore (leetcode.com/explore)

Difficulty: Medium |**Time:** 4 hours |**Deliverable:** Working code

Day 5: System Design Fundamentals

Morning Session (4 hrs)

- Requirements clarification
- Scalability patterns
- Caching strategies
- Load balancing
- Database design
- CAP theorem

Afternoon: Pick 1 Project

- URL shortener (TinyURL)
- Real-time chat system
- API rate limiter
- Real-time leaderboard
- Notification system

Resources (Free):

- System Design Handbook (algomaster.io)
- DesignGurus Guide (designgurus.io)
- InterviewBit (interviewbit.com)

Difficulty: Medium |**Time:** 4 hours |**Deliverable:** Working code

Week 1 Milestone Assessment

Learning Outcomes

- Python OOP mastery
- Advanced patterns
- DSA competency
- System thinking

Success Criteria

- 5 projects completed
- Core concepts understood
- Code quality ¿80%
- Ready for NLP phase

Assessment Activities:

- Code review of all 5 projects
- Technical Q&A on concepts
- System design explanation
- Readiness verification

Assessment Rubric:

- Functionality (50 pts) - All features working
- Code Quality (20 pts) - Clean, documented
- Design Principles (20 pts) - Architecture
- Presentation (10 pts) - Documentation quality

Day 6: NLP Basics with spaCy

Morning Session (4 hrs)

- NLP fundamentals
- spaCy pipeline
- Tokenization
- POS tagging
- Named Entity Recognition

Afternoon: Pick 1 Project

- Text tokenizer
- NER system
- Text similarity
- Dependency parser
- Entity linker

Resources (Free):

- spaCy Advanced Course (course.spacy.io)
- Real Python spaCy (realpython.com/spacy)
- GeeksforGeeks NLP (geeksforgeeks.org/nlp)

Day 7: NLP Processing & Text Analysis

Morning Session (4 hrs)

- Text preprocessing
- Lemmatization
- Stop word removal
- Text normalization
- TF-IDF vectorization
- Feature extraction

Afternoon: Pick 1 Project

- Preprocessing pipeline
- TF-IDF vectorizer
- Advanced text cleaner
- Keyword extractor
- Document summarizer

Resources (Free):

- Real Python Text Processing (realpython.com)
- NLTK Documentation (nltk.org)
- DataCamp Text Analytics (freemium)

Day 8: NLP Advanced Topics

Morning Session (4 hrs)

- Dependency parsing
- Syntax trees
- Coreference resolution
- Topic modeling
- Information extraction
- Semantic role labeling

Afternoon: Pick 1 Project

- Syntax parser
- Coreference resolution
- Relation extraction
- Semantic role labeler
- QA system (template-based)

Resources (Free):

- Advanced spaCy Course (course.spacy.io)
- Real Python NLP (realpython.com)
- GeeksforGeeks Advanced (geeksforgeeks.org/nlp)

Day 9: Machine Learning + Sentiment Analysis

Morning Session (4 hrs)

- Sentiment analysis approaches
- Feature engineering
- ML classifiers
- Model evaluation
- Imbalanced datasets
- Ensemble methods

Afternoon: Pick 1 Project

- Lexicon-based analyzer
- ML classifier
- Ensemble model
- Real-time analyzer
- Aspect-based sentiment

Resources (Free):

- DataCamp NLTK (datacamp.com/tutorial/nltk)
- Real Python Sentiment (realpython.com/sentiment)
- Towards Data Science (towardsdatascience.com)

Day 10: Word Embeddings

Morning Session (4 hrs)

- Word2Vec (CBOW, Skip-Gram)
- GloVe embeddings
- FastText
- Embedding visualization
- Pre-trained models
- Transfer learning

Afternoon: Pick 1 Project

- Word2Vec training
- Analogy solver
- Document similarity
- t-SNE visualization
- Semantic search engine

Resources (Free):

- GeeksforGeeks Word2Vec (geeksforgeeks.org)
- Milvus Embeddings (milvus.io)
- TensorFlow Word2Vec (tensorflow.org)

Week 2 Milestone Assessment

Learning Outcomes

- NLP pipelines
- ML classification
- Embeddings mastery
- Semantic search

Success Criteria

- NLP pipeline functional
- ML model ¿80% accuracy
- Embedding system works
- Ready for RAG phase

Assessment Activities:

- NLP pipeline demo
- Sentiment model evaluation
- Embedding visualization
- Component integration check

Day 11: RAG from Scratch

Morning Session (4 hrs)

- RAG architecture
- Vector stores
- Chunking strategies
- Retrieval mechanisms
- Re-ranking
- Integration patterns

Afternoon: Pick 1 Project

- Basic RAG system
- Document chunking
- Re-ranking retriever
- Full RAG pipeline
- Multi-doc RAG

Resources (Free):

- RAG from Scratch (linkedin.com/pulse)
- HuggingFace RAG (huggingface.co/blog)
- Glean RAG Guide (glean.com/blog)

Why RAG? Combines retrieval with generation for:

- Up-to-date information beyond training data
- Grounded, factual responses with citations
- Domain-specific knowledge integration

Key Components: Document Loading → Chunking → Embedding → Vector Store → Retrieval → LLM → Response

Day 12: Docling Document Parsing

Morning Session (4 hrs)

- Docling architecture
- PDF/DOCX/PPTX parsing
- Layout analysis
- OCR capabilities
- Table extraction
- Image extraction

Afternoon: Pick 1 Project

- Multi-format parser
- Table extractor
- Image extractor
- OCR processor
- Hierarchy extractor

Resources (Free):

- DataCamp Docling (datacamp.com/tutorial/docling)
- IBM Docling (github.com/docling-project)
- Geek Avenue (youtube.com - Docling Tutorial)

Day 13: LangChain RAG Implementation

Morning Session (4 hrs)

- LangChain framework
- Document loaders
- Text splitters
- Embeddings integration
- Vector stores
- LCEL expressions

Afternoon: Pick 1 Project

- PDF Q&A system
- Web scraping RAG
- Multi-index RAG
- Conversation RAG
- Advanced RAG

Resources (Free):

- LangChain RAG (python.langchain.com)
- Kody Simpson YouTube (youtube.com)
- DataCamp RAG (datacamp.com/courses)

Day 14: Streamlit UI Development

Morning Session (4 hrs)

- Streamlit basics
- Layout components
- Interactive widgets
- Data visualization
- File uploads
- State management

Afternoon: Pick 1 Project

- Interactive dashboard
- Chatbot interface
- Document uploader
- RAG interface
- Multi-page app

Resources (Free):

- Streamlit Docs (docs.streamlit.io)
- GeeksforGeeks Streamlit (geeksforgeeks.org)
- DataQuest Chatbot (dataquest.io)

Day 15: End-to-End Capstone Project

Morning Session (4 hrs)

- Architecture planning
- Component integration
- Testing strategies
- Performance optimization
- Error handling
- Deployment prep

Pick 1 Project (Full Stack)

- Document Intelligence
- Advanced Multi-Doc RAG
- Enterprise Knowledge Base
- Real-Time Chat Analyzer
- AI Research Tool

Integration Requirements:

- Docling: Multi-format document parsing
- LangChain: RAG orchestration & chains
- Streamlit: Interactive web interface
- All components working together

Capstone Evaluation (100 pts): Functionality (40) |Code Quality (30) |Architecture (20) |Documentation (10) **Minimum Score:** 70/100 to pass

Week 3 Milestone & Final Assessment

Learning Outcomes

- Complete RAG systems
- Document processing
- LLM orchestration
- Production applications

Success Criteria

- Capstone fully functional
- Code well-documented
- Architecture sound
- Production-ready

Production Readiness Checklist:

- Error handling implemented
- Performance optimized
- Documentation complete
- Deployment configuration ready

Final Assessment:

- Capstone project demonstration
- Architecture explanation
- Code walkthrough
- Future enhancements discussion

Congratulations! Next Steps

Technical Skills

- Advanced Python
- NLP & ML systems
- RAG architecture
- Full-stack development

Career Pathways

- AI/ML Engineer
- RAG Specialist
- LLM Infrastructure
- Product Engineer (AI)

Next Certifications:

- Azure AI Fundamentals (AI-900)
- LangChain Academy Certification

Continuous Learning Resources:

- LangChain Academy (academy.langchain.com)
- DeepLearning.AI Courses (deeplearning.ai)
- Hugging Face Course (huggingface.co/course)
- GitHub: Explore open source RAG projects

Keep Building, Keep Learning!