

Python for AI and Intelligent Systems

1. Python Foundations and Developer Environment

Development Environment - Python Installation (Windows, Mac, Ubuntu), Python Virtual Environments, Jupyter Notebooks, Google Colab, Package Installation using pip, Managing dependencies using requirements.txt

Python Basics - Introduction to Python, Python Syntax, Writing Python Programs, Variables and Constants, Data Types, Type Conversion, Comments and Escape Characters

Computing Fundamentals - Data on Computers, Binary Numbers, Characters and Strings, Numeric Representations

2 . Control Flow, Functions and Modular Programming

Decision Making - Boolean Expressions, Conditional Statements, Nested Decisions, Multi-way Decisions, Short Circuit Evaluation

Iteration - while loops, for loops, break, continue and pass

Functions - Creating Functions, Parameters and Return Values, Variable Scope, Lambda Functions, Recursive Functions

Modules and Packages - Import Statements, Creating Modules, Python Packages, Reusable Utility Libraries

3. Working with Data and Files

Strings - String Operations, String Manipulation, String Libraries

Files - Reading Files, Writing Files, Text Processing, File Paths

Structured Data - CSV Processing, JSON Processing, Configuration Files, Environment Variables

Data Structures - Lists, Tuples, Sets, Dictionaries, List and Dictionary Comprehensions

4. Python Advanced

Object Oriented Programming - Classes and Objects, Inheritance, Composition

Advanced Python - Iterators, Generators, Decorators, Context Managers

Modern Python - Type Hints, Dataclasses, Enumerations

Error Handling - Exceptions, Custom Exceptions, Retry Patterns, Defensive Programming

5. Python for APIs, Automation and Concurrency

Python Libraries - os, pathlib, datetime, random, math

Web Requests - REST APIs, HTTP Requests, Authentication Tokens, API Clients

Automation - Automation Scripts, Batch Processing, Scheduling Jobs

Concurrency - Threads, Multiprocessing, Async Programming, asyncio

Performance Considerations - Profiling, Measuring Execution Time, Optimization Basics

6. Python for Data Analysis and Machine Learning

NumPy - Arrays, Indexing, Slicing, Vectorized Operations, Matrix Operations

Pandas - DataFrames, Data Cleaning, Feature Engineering, Data Transformations

Data Visualization - Matplotlib, Exploratory Analysis

Machine Learning Foundations - Scikit-Learn, Model Training, Evaluation, Pipelines

Introduction to TensorFlow - Tensors, Training Workflow

7. Python for AI Systems

AI Application Architecture - AI Service Architecture, Batch vs Real-Time Inference, AI Microservices

Inference Systems - Model Serving, API Endpoints, Streaming Responses

Reliability Engineering - Logging, Monitoring, Evaluation, Guardrails, Observability

Packaging and Deployment - Docker Containers, Dependency, Management, Deployment Patterns

MLOps Foundations - CI/CD for ML, Model Lifecycle Management, Experiment Tracking

8. Python for Intelligent and Agentic Systems

Working with LLM APIs - API Clients, Prompt Templates, Structured Outputs

Embeddings and Retrieval - Embeddings, Vector Databases, Similarity Search

Retrieval Augmented Generation - Chunking, Retrieval Pipelines, Context Construction

Agentic AI - Tool Calling, Function Calling, Multi-Agent Workflows, Agent Memory

LLM Engineering - Prompt Design Patterns, Evaluation, Safety, Hallucination Detection

AWS for AI Engineering and Intelligent Systems

1. AWS Orientation, Account Setup, Cloud Concepts and Cost Safety

AWS Academy Cloud Foundations Mapping - Introduction to Cloud Computing, Advantages of the Cloud, Introduction to AWS, Moving to the AWS Cloud, AWS Global Infrastructure, AWS Services Overview, Cloud Economics and Billing, AWS Cloud Security

AWS account creation / login, AWS Free Tier awareness, Root user vs IAM user, Enable MFA, AWS Console navigation, AWS Regions and Availability Zones, Service search and dashboard navigation, Billing dashboard, Cost Explorer, AWS Budgets, Basic tagging and naming conventions, Resource cleanup awareness

Demonstration - Log in to AWS, Enable MFA, Navigate the AWS Console, Identify the selected AWS Region, Open EC2, S3, IAM, CloudWatch, Billing and SageMaker dashboards, Explore AWS CloudShell, Create a basic budget alert, Review where to check active resources and costs

2. Compute, Linux Basics and Application Hosting with EC2

AWS Academy Cloud Foundations Mapping - Compute Services Overview, Amazon EC2 Part 1, Amazon EC2 Part 2, Amazon EC2 Part 3, Console Demonstration - EC2, Lab - Introduction to Amazon EC2, EC2 Cost Optimization, Container Services, AWS Lambda

EC2 instance creation, AMI selection, Instance types, Key pairs, Security groups, Browser-based SSH / EC2 Instance Connect, Public IP and public DNS, Start, stop, reboot and terminate instances, Basic Linux commands, Installing Python, pip, Git and virtual environments, Running a simple Flask application

Demonstration: Launch EC2 and Host a Simple Python Web App - Launch an Ubuntu EC2 instance, Connect using browser-based SSH, Run system update, commands: sudo apt update, sudo apt upgrade, Install Python, pip and Git, Create a simple Flask application, Run the application on a chosen port, Configure security group rules, Access the application from a browser, Stop and restart the instance, Terminate the instance when finished

3. Networking, VPC, Security Groups and API Access

AWS Academy Cloud Foundations Mapping - Networking Basics, Amazon VPC, Console Demonstration - VPC Wizard, VPC Networking, VPC Security, Route 53, CloudFront, Lab - Build your VPC and Launch a Web Server

Default VPC, Custom VPC overview, Public subnet vs private subnet, Internet Gateway, Route table basics, Security groups as virtual firewalls, Common ports: 22, 80, 443, 5000, 8000, 8080, API access through browser and API clients, Basic troubleshooting for unreachable applications, Introduction to API Gateway

Demonstration: Make an EC2-Hosted API Browser Accessible - Review the default VPC, Launch EC2 into a public subnet, Configure security groups, Run a Flask API, Access the API using public IP, Test different ports, Observe the effect of opening and closing ports, Discuss safe access practices

4. Storage, Data Lakes and Dataset Access with S3, EBS and Athena

AWS Academy Cloud Foundations Mapping - AWS EBS, Console Demonstration - EBS, Lab - Working with EBS, AWS S3, Console Demonstration - S3, AWS EFS, S3 Glacier

S3 bucket creation, Bucket naming rules, Public vs private access, Upload and download objects, S3 folders/prefixes, Bucket properties, S3 versioning overview, Reading S3 files from Python, Dataset storage patterns, EBS as instance storage, Introduction to Athena for querying data in S3

Demonstration: Store and Read an AI Dataset from S3 - Create an S3 bucket, Upload a CSV dataset, Upload JSON configuration files, Upload sample text or image files, Read the dataset from a Python script or notebook, Write output files back to S3, Explore object permissions, Delete objects and buckets after use

5. Data Engineering for AI with AWS Glue, Data Catalog and Athena

AWS Academy Cloud Foundations Mapping - Databases, Storage Review, Cloud Architecture

Why data preparation matters for AI and ML, ETL concepts, Data cleaning and transformation, AWS Glue overview, Glue Crawlers, Glue Data Catalog, Glue Jobs, Schema discovery, CSV to Parquet conversion, Querying transformed data using Athena, Data quality checks introduction

Demonstration: Build a Simple Data Preparation Pipeline - Upload raw CSV data to S3, Create a Glue Crawler, Catalog the dataset, View schema in Glue Data Catalog, Query the data using Athena, Create a simple Glue ETL job, Clean or transform selected columns, Save transformed data back to S3, Query transformed data using Athena

6. Machine Learning Development with Amazon SageMaker AI

AWS Academy Cloud Foundations Mapping - None, Compute Review, Storage Review, Cloud Architecture, Databases

SageMaker AI overview, SageMaker Studio / notebook environment overview, Notebook instances or managed notebook environment, S3 as training data source, Training jobs concept, Model artifacts, Basic experiment tracking concept, Model deployment overview, Endpoint cost awareness, Difference between notebook experimentation and production deployment

Demonstration: Train a Simple ML Model Using SageMaker: Open SageMaker environment, Create or access a notebook, Load dataset from S3, Train a simple model, Save model artifact to S3, Review logs and outputs, Understand where training data, code and model artifacts are stored

7. MLOps Foundations: Containers, Model Deployment, Pipelines and Monitoring

AWS Academy Cloud Foundations Mapping - Less, Container Services, AWS Lambda, Auto Scaling and Monitoring, Cloud Architecture, Compute Review

Why MLOps is required, Model lifecycle overview, Model packaging, Docker concepts, Amazon ECR, ECS introduction, Lambda for lightweight functions, Step Functions for workflow orchestration, SageMaker Pipelines overview, CI/CD concepts, CloudWatch metrics and logs, Model monitoring introduction, Secrets Manager introduction

Demonstration: Package and Monitor a Simple Inference Service - Containerize a simple Flask inference app locally, Understand how images are pushed to ECR, Review ECS as a container runtime option, Create a simple Lambda function, Create a basic Step Functions workflow, Trigger a simple workflow step, View logs in CloudWatch, Store a configuration value or secret securely

8. Generative AI and Agentic AI on AWS

AWS Academy Cloud Foundations Mapping - AWS Academy Topic, Suggested Coverage, Cloud Architecture, Security Review, Monitoring Review, Compute Review, Storage Review

Amazon Bedrock overview, Foundation models on AWS, Prompt testing, Text generation use cases, Embeddings overview, RAG architecture overview, Amazon Bedrock Knowledge Bases introduction, Amazon Bedrock Agents introduction, Guardrails concept, Lambda as a tool/action layer, Step Functions for agent workflow orchestration, OpenSearch Serverless introduction for vector search, DynamoDB introduction for application state/memory, API Gateway for exposing AI services, CloudWatch for GenAI app monitoring, Secrets Manager for API keys and credentials

Demonstration: Build a Simple Generative AI Workflow

Demonstration 1: Prompt-Based GenAI App - Use Amazon Bedrock to test a prompt, Send input text, Generate a response, Store input/output in S3 or DynamoDB, View logs in CloudWatch

Demonstration 2: Simple RAG-Style Flow, Store documents in S3, Understand embeddings and retrieval conceptually, Explore Bedrock Knowledge Bases or OpenSearch Serverless conceptually, Generate a grounded response

Demonstration 3: Simple Agentic Workflow - Use Lambda as a tool/action, Use Step Functions to orchestrate steps, Connect the flow to a Bedrock-based reasoning step conceptually, Log execution results