



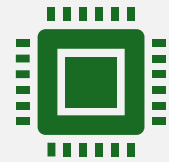
# Advanced Python

Surendra Panpaliya

# Week1 ( Mon, Tue, Thurs)



Day 1: Python Recap +  
Environment & Tooling



Day 2: Functional Programming &  
Object-Oriented Design



Day 3: Advanced Python  
Concepts

# Week2 ( Mon, Tue, Wed, Thurs)



Day 4: Concurrency and Async Programming



Day 5: Web Services with FastAPI



Day 6: Azure Functions & Cloud Deployment



Day 7: Testing, Linting & Final Project

# ***Day 1: Python Recap + Environment & Tooling***

Quick Python vs C# syntax mapping

Variables, data types, control flow in Python

Comprehensions (List, Dict, Set)

Functions: \*args, \*\*kwargs, lambda

Tooling: pip, venv, poetry, dependency locking

# ***Day 1: Python Recap + Environment & Tooling***



Hands-On Lab:



Set up Python project & create utility functions



*Python vs C#: Static typing vs dynamic,*



*Main method vs scripts,*



*No semicolons/curly braces*

# ***Day 2: Functional Programming & Object- Oriented Design***



Pythonic FP: map, filter, reduce, functools



Object-Oriented Programming:



C# vs Python class structure



SOLID principles and Mixins



Design Patterns: Singleton, Factory, Strategy

# ***Day 2: Functional Programming & Object- Oriented Design***

Hands-On Lab:

Create modular design for plugin architecture

Authentication using OAuth / Azure AD

*Python vs C#: Interfaces vs duck typing,*

*Access modifiers*

***Day 3:  
Advanced  
Python  
Concepts***

Decorators: Logging, validation, chaining

Context Managers: with statement, `__enter__`, `__exit__`

Generators and yield, pipelines

Metaclasses: Framework-level magic



***Day 3:  
Advanced  
Python  
Concepts***

Hands-On Lab:

Logger with decorators  
and context managers

*C# Attributes vs Python  
Decorators*

***Day 4:  
Concurrency  
and Async  
Programming***

Concurrency: threading vs multiprocessing

Async IO: asyncio, event loop, aiohttp

Profiling Tools: cProfile,

line\_profiler, memory\_profiler

# ***Day 4: Concurrency and Async Programming***



Hands-On Lab:



Build async scraper with aiohttp



*C# async/await vs Python  
async/await*

# ***Day 5: Web Services with FastAPI***



FastAPI project structure vs ASP.NET Core WebAPI



Path/query parameters, request validation



Dependency Injection (DI)



Serving with Uvicorn + Gunicorn

# ***Day 5: Web Services with FastAPI***

Hands-On Lab:

Build CRUD APIs using FastAPI

*ASP.NET Route attributes vs FastAPI decorators*

# ***Day 6: Azure Functions & Cloud Deployment***

Azure Functions for Python:

Local setup and deployment

Comparing Dockerized App vs Azure Function

CI/CD in Azure Pipelines / GitHub Actions

***Day 6: Azure  
Functions &  
Cloud  
Deployment***

Hands-On Lab:

Convert API to Azure  
Function and deploy

*C# Azure Functions vs  
Python Azure Functions*

# ***Day 7: Testing, Linting & Final Project***



Code Quality & Linting:



mypy, ruff, black, flake8, pre-commit



Testing: pytest, mocking, assertions



Git Best Practices: PRs, reviews, branching strategy



# ***Day 7: Testing, Linting & Final Project***

Final Capstone Project:

Build Azure Function or FastAPI-based application

Apply Clean Architecture & CQRS

Present project

*C# unit test framework vs pytest*

# Software Installation Requirements

**Download Link:Anaconda:**

<https://www.anaconda.com/products/distribution>

**Visual Studio Code :**

<https://code.visualstudio.com/Download>

# Software Installation Requirements

**Azure CLI Version: 2.50.0+**

<https://docs.microsoft.com/cli/azure/install-azure-cli>

**Azure Functions Tools:**

<https://docs.microsoft.com/azure/azure-functions/functions-run-local#v2>

**Python  
Libraries  
(installed via  
pip or Conda)**

FastAPI ( $\geq 0.95.0$ )

aiohttp ( $\geq 3.8.0$ )

requests ( $\geq 2.31.0$ )

pandas ( $\geq 2.0.0$ )

pytest ( $\geq 7.0.0$ )

uvicorn ( $\geq 0.22.0$ )

Happy Learning@!!  
Thanks for Your  
Patience 😊

Surendra Panpaliya  
GKTCS Innovations

