

Title: Advanced Python 3 for IC Design Engineers

Course Description

A three-day Advanced Python course for design, evaluation, testing or production support engineers with existing Python experience, who wish to deepen their programming knowledge further. Participants will be enabled to design robust Python applications using modern development methods, including how to use an AI Assistant to improve efficiency when writing code and adding AI features to Python applications.

Who is the course for?

Engineers in a design, evaluation, testing or production support role with existing Python experience, who wish to leverage the power of Python further. It is expected that attendees have some work experience with Python and/or have attended the Introductory Python course.

Learning Outcomes:

At the conclusion of this course, attendees will be able to:

- Design robust Python applications using modern development methods
- Use Classes to maximise code reuse and portability
- Use design patterns when creating Python applications
- Design and create multi-threaded Python applications
- Create standalone Python interface with Tkinter and PyQt
- Work with JSON and XML data
- Access HTML from web pages
- Create web based applications with Flask
- Access and Edit Data from a relational database
- Create and run Unit tests
- Use Numpy for numerical calculations
- Use Pandas for data analysis
- Create your own Python modules
- Use an Al Assistant to improve efficiency when writing code
- Add AI features to a Python application

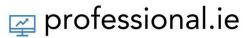
Course materials:

All participants will receive course example files, cheat-sheets and PDF reference materials prior to the course or on the first morning. Specifically, they will receive:

- Advanced Python Textbook
- GitHub repository for all course materials
- Exercises with Solutions
- Cheat Sheets for handy reference
- Data files for analytics problems and demos
- Online textbook with exercises and solutions for data science topics
- Online textbook with exercises and solutions for data visualisation topics

Software Environment:

Python with Visual Studio Code with Python and Jupyter Notebooks extensions. PyCharm, Spyder and other Python development tools and environments like Anaconda can be used if required by participants.



Labs and Exercises

This is a practical hands-on course where participants write, test and debug code throughout. Exercises and examples are tailored to the practical needs and backgrounds of electronic engineers. Lab exercises are completed by participants and overall about 30% of the course time is used for hands-on exercises and 70% for demonstration.



Day 1 – Advanced Python 3

Python Review

Data Types and Variables
Flow of Control
Functions
Lists, Tuples and Dictionaries
Files
Exceptions

Classes

Class variables and methods
Working with Properties
Special Class methods - "dunder" methods
The Python data model
Writing classes that can be sorted and iterated

Decorators

What is a decorator?

Some sample decorators

Decorators for oo programming @property @dataclass

Creating your own decorators

Test Driven Development

Test Driven Development – Unit Testing
Pytest
VSCode Pytest integration
Writing a test fixture to run for every test
Writing a test fixture to run for every session

Database Programming

The SQLite module
Database CRUD methods
The Data Access Object design pattern
Adding unit tests to the DAO

UI Programming

The TKInter module
A basic windowed application
Widgets
Adding a list
Event handlers

Using AI Assistants for Coding

Overview of Al Warnings Prompt engineering techniques Getting ChatGPT or Copilot to write code Iterative development



Day 2 - Advanced Python 3

Working with APIs

The requests module
Making a call
Processing JSON / XML data
Blocking vs asynchronous calls
async await

PyQT UI Development

The PyQT module
PyQT vs TKInter
Creating a PyQT ui application
Adding widgets
Events
Listbox
Asynchronous api call from PyQT
Adding extra data to a ListBoxItem

Functional Programming

Functions are first class objects Passing and returning functions Lambda Functions Map Filter Reduce

Web scraping

The BeautifulSoup module
Basic concepts of web scraping
Extracting all elements of a particular type
Filtering out unwanted tags
Extracting elements from tags
Sample application, extract information from worldometers.info

Regular Expressions

Creating expressions
Testing your expressions using regex101.com
Searching
Using groups in an expression
Replace
Sample - command line utility to obfuscate email addresses



Day 3 - Advanced Python 3

Writing and Maintaining your Own Python Library

Write your own iterators, generators and decorators
The __init__.py file
.toml files
Installing your module for local development
Adding an external dependency
Building your module

Introduction To Web Application Programming

The Flask module
Web development basics
Endpoints
Returning HTML
Using templates - the jinja template engine
Deploying your flask application

Adding AI To Your Python Application Using ChatGPT API

Making an API call from Python Introducing the ChatGPT API Making a request A sample AI powered application

Understanding Git and GitHub

Code for course is shared using GitHub Cloning a repository Pushing to a repo Some GitHub features Configuring a GitHub action to run automated tests

Python for Data Science - Numpy and Pandas

The Data Science Handbook Clone repository Installing dependencies from requirements.txt Review Jupyter Notebooks Execute all code

Python For Data Visualisation - MatplotLib and Seaborn

The matplotlib samples page Copying and running a sample dataforsci.com Cloning repository Installing dependencies Review topics

Executing code