



Geometric Progression

We Bring AI to Reality.

Learn Python, Vibe Coding & Py for Excel:

With a focus on Financial Markets & Business applications

Overview

Welcome to our Python course, a transformative programme designed to equip you with the Python programming skills essential for today's business world starting from scratch. Python is now incorporated into Excel and is becoming an indispensable tool for financial analysis & modelling. This course addresses the urgent need for proficiency in this versatile language.

Our curriculum is meticulously tailored to fast-track your learning, blending core Python programming skills with specific applications in business and the financial markets. From understanding basic data types and control structures to mastering data manipulation with Numpy and Pandas.



What sets our course apart is not just the content but the format. We combine the best of in-person and online learning, ensuring you're not just passively watching pre-recorded lectures but actively engaging with instructors and peers. This interactive approach is invaluable, especially when starting out, as it provides immediate feedback, personalized support, and the dynamic exchange of ideas and questions with fellow participants. The course will also be recorded to get the best of both experiences.

Moreover, we accelerate your learning with AI copilots such as ChatGPT and Gemini, wrapped inside our signature **Vibe Coding** workflow—pairing you with an always-on AI teammate and live instructor guidance. Because the course takes you **inside Excel to run native Python code**, you'll learn how to super-charge the spreadsheets you already know, putting full-scale programming power at your fingertips and keeping you a step ahead of the curve.

Additional Features:

- Choose from a host of optional modules.
- Includes Lifetime access to comprehensive electronic course materials, including insightful videos and a curated list of resources for deeper exploration & Cheat Sheets.
- Recorded sessions for future reference.

[Please refer to the detailed agenda below for more information.](#)

Level	Introduction to Intermediate.
Duration	Customized to your needs.
Requirements	Computer with internet access. Contact us for files to download and setup.
Suitable For	Anyone who wants to learn to program, automate tasks, or learn Python.
Not Suitable For	Anyone who does not understand the basics of using a computer.



Geometric Progression

We Bring AI to Reality.

Tutor: Mark Raffaelli CFA, FRM

Mark obtained his **CFA Charter** in 2000 and became a **Fellow of the Global Association of Risk Professionals (FRM)** in 2001. His experience spans:

- **Trading spot and derivative products** professionally.
- **Developing quantitative financial models** for surveillance, performance attribution, price validation, pricing, risk and automation.
- **Building software and apps** for the investment and insurance industries.
- **Applying machine and deep-learning techniques** (scikit-learn, TensorFlow, PyTorch) to capital-markets problems for banks, asset managers and corporates.
- **Performing time-series analysis, regime-shift detection and optimisation** - with and without AI.
- **Designing next-generation AI solutions:** agent-based workflows, retrieval-augmented-generation (RAG) pipelines and fine-tuned large-language-model tools - built with frameworks and Model Context Protocol (MCP) - so clients can move from proof-of-concept to production quickly.
- **Implementing enterprise-scale AI strategies** that turn ideas into measurable value.

Mark has also contributed to the South African bond market, writing pricing and curve specifications for Floating-Rate Notes (FRNs) and Inflation-Linked Bonds (ILBs) - standards still used today - and serving on the Quant Committee for indices and curves.

Known for his engaging style, Mark breaks down technical ideas, removes the jargon and leaves attendees confident to apply what they've learned.

Things delegates have said about our Python training:

- *"This is the best training I have attended in my professional career. We had a short course while I worked at a previous firm on Python but I felt I learnt more from Mark in the first 3 hours alone than I learnt on the course we had."*
- *"Mark's Passion for programming is inspiring."*
- *"Thanks Mark, I now know how to automate the boring stuff out of my job"*
- *"I loved the way we learnt by programming exercises ourselves."*
- *"I'd recommend my colleagues go out of their way to make time for this. Invaluable and brilliantly presented! "*
- *"5 out of 5 stars from me, anyone wanting to learn Python should seriously attend this course. Fantastic!"*
- *"Thoroughly enjoyed it. I feel empowered. Thank you."*

Organisations represented at our previous public courses include:





Detailed Agenda

1. The stuff you need to know before we start.

Delegates will be able to:

- Understand why it is so important to know how to Program in today's world.
- Distinguish between Modelling versus professional programming.
- Navigate and efficiently use Jupyter notebooks.
- Create and use notebooks.
- Create comments in notebook cells.
- Create markup cells to create a rich user experience incorporating markup language for headings, paragraphs, images, links etc.

2. Datatypes

Delegates will be able to create, use and explain of the different types of datatypes. Delegates will be able to:

- Use the print statement and the "type" statement.
- Create a string and output string.
- Output a string using a host of different string formatting methods.
- Create, use and explain integers.
- Create, use and explain floating type numbers.
- Create code to cast from one datatype to another.
- Create and use variables.

3. The world of Functions

Delegates will be able to:

- Create a function.
- Understand and correctly indent code to make the function work correctly.
- Understand and use white-space appropriately.
- Define the function Signature.
- Create arguments and parameters and understand the difference.
- Create and use "docstrings".
- Create user-defined functions.
- Create, use and understand default arguments and optional arguments.
- Create and implement python built in functions.





4. Scope

Delegates will be able to:

- Understand and define the difference between local and global scope.
- Understand the concept of enclosing scope and variable clash.

5. Conditional logic

Delegates will be able to:

- Create an if statement and use the else-if and else Logic.
- Put together some nested if statements.
- Understand the concept of XOR

6. Loops

Delegates will be able to:

- Define, create and use the range function.
- Create a for loop and understand the logic of how the code is processed.
- Create a while loop and understand the concept of incrementation.
- Create nested loops.

7. Useful Loop Functions

Delegates will be able to:

- Solve algorithmic problems using flags.
- Implement the “break” out of a loop code.
- Implement the “continue” and loop code.
- Use and understand the “pass” place-holder.
- Understand the concept of the “iter” method and implement it.
- Understanding the “yield” statement.

8. Lists

Delegates will be able to:

- Use and create lists.
- Slice a list.
- Append, delete and extend lists.
- Pop an item off a list.
- Loop through items of a list.
- Implement list functions.





9. Sets & Tuples

Delegates will be able to:

- Create a set and implement set operations.
- Use a set to derive unique items.
- Use set mathematics to filter data.
- Create a tuple.
- Implement tuple functions.

10. Dictionaries

Delegates will be able to:

- Create, use and develop a dictionary.
- Use economic stats to fill a dictionary.
- Slice a dictionary.
- Implement a dictionary and list comprehension.
- Loop through the dictionary.

11. Modules

Delegates will be able to:

- Understand and perform a pip install.
- Import external modules.
- Import self-created modules.
- Create a “namespace”.
- Import modules with wild cards.
- Create a module.

12. Code Editors

Delegates will be able to:

- Evaluate different code editors such as, VS Code, Jupyter, Cursor, WindSurf, Pycharm, Spyder etc.
- Independently set up an environment.
- Download a python extension.
- Select a python interpreter.
- Implement debugging such as step through, step over and break points.





13. Object Orientated Programming (OOP)

Delegates will be able to:

- Explain the difference between procedural code and object orientated code.
- Create a class.
- Encapsulate functions inside a class.
- Create an object constructor.
- Create an object instance.
- Call an instance attributes and methods.
- Implement dunder methods.
- Using "self" in code.
- Build their own OOP Calculator.

14. Functional Programming and Lambda Functions

Delegates will be able to:

- Understand and use a first class function.
- Pass a function as an argument.
- Create a lambda function.
- Implement the "filter", "map", "reduce" statement on a lambda function.

15. Mastering "Vibe Coding": Programming with your AI Teammate

You will partner with AI copilots like ChatGPT or Google Gemini etc under live instructor guidance. We will dedicate time to ensure all participants can:

- Effectively prompt and interact with AI assistants for Python programming tasks.
- Leverage AI for code generation, debugging, and understanding complex concepts.
- Integrate AI assistance seamlessly into their development process to accelerate learning and problem-solving.
- Understand best practices for using AI tools responsibly and efficiently in a programming context.

16. Wrapping up with Environments, Zipping, widgets, decorators.

Delegates will be able to:

- Use Jupyter widgets.
- Use the Python Zip function.
- Understand and use **@Decorators**.
- Implement and use the **Datetime module**.
- Implement and use the DateUtil module.
- Create a **graphical user interface (GUI)**.
- Create an exe using pyinstaller.
- Setting up a Python Environment.





Core Python Libraries for Analysts or anyone doing Analysis / Analytics.

17. Numpy.

Delegates will be able to:

- Convert different datatypes to numpy arrays.
- Measure Array shapes, dimensions and size.
- Use vectors in numpy.
- Numpy math.
- Portray images as vectors.
- Reshape, ravel and flatten arrays.
- Implement Numpy functions and seeding.

18. Pandas.

Delegates will be able to:

- Convert datatypes to a pandas series
- Create, update and delete a series.
- Implement Series functions and filter a series.
- Convert datatypes to a pandas dataframe.
- Create, update and delete a dataframe.
- Implement dataframe functions.
- Distinguish between the commands “display vs print”.
- Implement different reference conventions (loc and iloc etc).
- Perform selection, filtering and assignment with loc, iloc, at,iat.

19. Intermediate Pandas.

Delegates will be able to:

- Loop through big data with Pandas.
- Use superior alternatives to loops with referencing with Pandas.
- Import data from, csv, Excel and pdf into Pandas.
- Perform HTML Webscraping with Pandas.
- Create code for remote access of data with Pandas
- Implement Pandas TimeSeries and datetime.
- Build an algo trader and backtest the results.
- Perform data interpolation with pandas.
- Output to Excel with pandas.





20. Plotting Charts with Python.

Delegates will be able to:

- Chart with Matplotlib.
- Plot points, scatterplots and a line.
- Add some labels and features to the graph with code.
- Plot various charts with code.
- Create multiple plots with subplots.
- Create variations of subplots.
- Add axes manually in code.
- Use formulas to plot with numpy
- Create 3D Plots.
- Rotate the plots and Azimuth's.
- Create quick plots from pandas to matplotlib
- Add brief tour of other charting modules.

21. Python Power in Excel

Delegates will be able to:

- Understand the integration of Python within Microsoft Excel.
- Activate and utilize Python directly within Excel cells (PY cells).
- Reference Excel data (ranges, tables, Power Query connections) within Python code using standard Excel formulas or Python functions.
- Leverage core Python libraries, such as Pandas and Matplotlib, for advanced data manipulation, analysis, and visualization directly within an Excel environment.
- Output Python results, including DataFrames and charts, back into the Excel grid or as objects.
- Explore practical use cases for financial modelling, data cleaning, and task automation using Python in Excel.
- Recognize the benefits and current limitations of using Python within Excel.

22. A Simple Introduction to Portfolios and Algo Trading with Python

- PyPortfolio and other Portfolio libraries.
 - Optimizing a portfolio.
- Algo trading and Algo trading libraries.
 - A simple Algo Trade
- Valuation libraries.

The purpose of this section is to show just how easy it is to take advantage of existing modules.





Geometric Progression

We Bring AI to Reality.

Optional Modules

Please note that all the modules listed below have detailed agendas and can be customized to your specific needs. Please contact us for more information or customization. The focus of many of the applications will be centered around portfolios, trading, risk and financial markets unless directed otherwise.

- **Automation:** Learn to automate repetitive tasks and workflows using Python scripts.
- **Webscraping:** Extract data from websites for analysis and other applications.
- **Mathematics with Python:** Explore advanced mathematical computations and libraries.
- **Statistics with Python:** Perform statistical analysis and modelling.
- **Optimization with Python:** Apply optimization techniques to solve complex problems.
- **Algo Trading:** Develop and back-test algorithmic trading strategies.
- **Portfolio Management:** Utilize Python for portfolio construction, analysis, and management.

We also offer training on pretty much all aspect of AI from modelling, to implementing traditional machine learning and deep learning algorithms, LLMs and agentic workflows.

Contact us to tailor this course to your team's specific needs.

