

**Advanced Python for Financial Technologies**

**Course Number:** PYTH-134
**Duration:** 3 days

**Overview**

Maximize returns. Visualize your portfolio. Execute your latest Killer Trading Algorithm. All of these and more are easily within reach by harnessing the open-source power of Python.

This Advanced Python course, Python for Finance Training, teaches you how to apply Python to a diverse range of financial technology applications, including acquiring data from popular financial data providers, as well as cleaning, exploring, and visualizing the resulting datasets. Attendees learn how to approach the implementation of algorithmic models and how to construct rich and insightful models, with an emphasis on ethics, compliance, and security.

**Prerequisites**

All Python For Finance training students should already be somewhat familiar with fundamental Python syntax and concepts.

**Materials**

All students receive comprehensive courseware.

**Software Needed on Each Student PC**

* Any Windows, Linux, or macOS operating system
* Python 3.x installed (Anaconda bundle recommended)
* An IDE with Python support (Jupyter Notebook, Spyder or PyCharm Community Edition)

**Objectives**

* Automatically extract financial data from common data providers
* Know how to clean, aggregate, and manipulate financial data effectively
* Conduct elementary time series analysis
* Understand stochastic processes and common noise models
* Construct models for inference and forecasting, such as ARIMA and linear and logistic regression
* Generate powerful visualizations, such as candlestick charts
* Extract financial data by scraping websites
* Understand the fundamentals of supervised and unsupervised machine learning models as applied to finance
* Apply Recurrent Neural Nets (RNNs) and Long Short-Term Memory Units (LSTMs) to financial time series and understand their limitations
* Understand the principles behind Blockchain technology

**Outline**

* Introduction
* Crunching the Numbers: Numerical Python With NumPy
	+ Introduction to the n-d-array
	+ NumPy operations
	+ Broadcasting
	+ Missing data in NumPy (masked array)
	+ NumPy structured arrays
	+ Improving performance through vectorization
	+ Random number generation
	+ Introduction to Monte-Carlo methods
	+ General approaches to implementing mathematical algorithms
* Acquiring and Manipulating Financial Data With Pandas and Pandas-Datareader
	+ Series versus DataFrames
	+ Overview of data types in pandas
	+ Pandas I/O tools: CSV/Excel/SQL
	+ Pandas I/O tools: Pandas-datareader
	+ Subsetting DataFrames
	+ Creating and deleting variables
	+ Discretization of continuous data
	+ Scaling and standardizing data
	+ Identifying duplicates
	+ Dummy coding
* Exploratory Data Analysis and Advanced Pandas Methods
	+ Uni- and multivariate statistical summaries and detecting outliers
	+ Group-wise calculations using pandas
	+ Pivot tables
	+ Long to wide and back: pivoting, stacking and melting
	+ Python visualization: Matplotlib and seaborn
	+ Pandas visualization: histograms, bar and box plots
	+ Pandas visualization: Scatter plots and pie charts
	+ Group-by plotting
	+ Pandas plot formatting
	+ mpl-finance and candlestick charts
	+ Merging DataFrames
	+ Pandas string methods
	+ Implementing regular expressions in pandas
	+ Handling missing data in pandas
* Elementary Time Series Analysis
	+ Date/time formats in Python and pandas
	+ Running/rolling aggregates
	+ Resampling
* Stochastic Processes
	+ Overview of noise models
	+ Stationarity
	+ Random walks and martingales
	+ Brownian motion
	+ Diffusion models
	+ The Black-Scholes model—and its limitations
* Time Series Forecasting
	+ De-trending and seasonality
	+ Interpolation and extrapolation
	+ Auto-Regressive Integrated Moving Average (ARIMA) models
* Measuring Impact: Testing For Group Differences
	+ Null hypothesis testing and p-values
	+ Group comparisons (p-values, t-tests, ANOVA, Chi-square tests)
	+ Correlation
* Progressing, With Regression Models
	+ Linear regression
	+ Logistic regression
	+ Regression on count outcomes (Poisson processes)
* Conclusion
* Optional: Machine Learning Fundamentals for Finance with scikit-learn
	+ Requirements: NumPy, pandas. Time required: 4 hours
	+ Machine learning approaches to multivariate statistics
	+ Machine Learning theory
	+ Data pre-processing
	+ Supervised versus Unsupervised learning
	+ Unsupervised learning: clustering
		- Clustering algorithms
		- Evaluating cluster performance
	+ Dimensionality reduction
		- A priori
		- Principal component analysis (PCA)
		- Penalized regression
	+ Supervised learning: regression
		- Linear regression
		- Penalized linear regression
		- Stochastic gradient descent
		- Scoring new data sets
		- Cross-validation
		- Variance-bias trade-off
		- Feature importance
	+ Supervised learning: classification
		- Logistic regression
		- LASSO
		- Random forests
		- Ensemble methods
		- Feature importance
		- Scoring new data sets
		- Cross-validation
* Optional: Recurrent Neural Nets and LSTMs with PyTorch
	+ Requirements: NumPy, pandas, Machine Learning fundamentals. Time required: 4 hours
	+ Introduction to PyTorch
		- Introduction to tensor algebra and calculus
		- Tensor algebra in PyTorch
		- Training and validating models
	+ Regression in PyTorch
		- Optimizers in PyTorch
		- Linear regression
		- Logistic regression
	+ Artificial Neural Networks
		- Overview of Artificial Neural Networks (ANNs)
		- Recurrent Neural Networks (RNNs)
		- Sequence models and Long Short-Term Memory Networks (LSTMs)
	+ RNNs/LSTMs with PyTorch
		- Building, training and validating a basic ANN
		- Creating a RNN
		- Building a LSTM
		- Applications to financial time series, and cautionary tales
* Optional: Scraping By: Obtaining Financial Data from Publicly Accessible Websites
	+ Requirements: Base Python. Time required: 2 hours
	+ Parsing HTML/CSS with BeautifulSoup
		- Navigating tree data structures
		- Selecting named node elements
		- Selecting by property
	+ Establishing a Connection
		- Urllib3 and connections
		- POST and GET directives
	+ Building a Web Scraper
		- Parsing a list of websites
		- Collecting and storing data
	+ Advanced Scraping: Building a Web Spider with Scrapy
* Optional: Blockchain technologies
	+ Requirements: Basic Python, NumPy (useful, but not mandatory). Time required: 4 hours.
	+ The Ingredients For a Blockchain
		- Transaction records
		- The distributed ledger
		- Chain validation
		- Nonces
	+ The Hash Function
		- Overview of hash functions and tables
		- Cryptographic hash functions
		- Proof-of-work
	+ Advanced Functions
		- Return statements
		- The JSON format
		- Exception trapping
		- Assertions
	+ Constructing Your Own Blockchain
		- Generating a block
		- The genesis block
		- Generating a chain though block validation
	+ Shortcomings of current blockchain technologies