

# Python for Finance

## Why Python?

The answer is simple: it is powerful yet very accessible. Python has become the most popular programming language for data science because it allows us to forget about the tedious parts of programming and offers us an environment where we can quickly jot down our ideas and put concepts directly into action.

### 1. Installing Python 3.9.8 (or higher) and packages from the Python Package Index

Python is available for all three major operating systems - Microsoft Windows, mac OS, and Linux - and the installer, as well as the documentation, can be downloaded from the official Python website: <https://www.python.org/downloads/>. It is recommended you use the most recent version of Python 3 that is currently available.

The additional packages can be installed via the pip installer program, which has been part of the Python standard library since Python 3.3. More information about pip can be found at <https://docs.python.org/3/installing/index.html>.

After we have successfully installed Python, we can execute pip from the **open command prompt** [windows + X /run] to install additional Python packages:

```
C:\Users\user\AppData\...\Python\Python39\python -m pip install package_name
```

**package\_name** = matplotlib, yfinance, pandas, pandas-datareader, numpy, beautifulsoup4, scikit-learn, scipy, seaborn

Already installed packages can be updated via the --upgrade flag:

```
C:\Users\user\AppData\...\Python\Python39\python -m pip install --upgrade package_name
```

### 2. Using the Anaconda Python distribution and package manager

A highly recommended alternative Python distribution for scientific computing is Anaconda by Continuum Analytics. Anaconda is a free-including for commercial use-enterprise-ready Python distribution that bundles all the essential Python packages for data science, math, and engineering in one user-friendly cross-platform distribution. The Anaconda installer can be downloaded at <https://www.anaconda.com/products/individual> , and an Anaconda quick-start guide is available at <https://conda.io/docs/test-drive.html>.

After successfully installing Anaconda, we can install new Python packages from the **open command prompt** [windows + X /run] or **Anaconda prompt** to install additional Python packages, using the following command:

```
C:/.../Anaconda3/scripts pip install package_name (open command prompt)
```

```
conda install package_name (Anaconda prompt)
```

**package\_name** = matplotlib, yfinance, pandas, pandas-datareader, numpy, beautifulsoup4, scikit-learn, scipy, seaborn

Already installed packages can be updated via the --upgrade flag:

```
C:/Anaconda3/scripts pip install package_name --upgrade (open command prompt)
```

```
conda upgrade package_name (Anaconda prompt)
```

## Web Resources:

<https://www.python.org/doc/>

<http://wiki.python.org/moin/BeginnersGuide/>

## For HELP see:

<https://stackoverflow.com/>

## Courses in Coursera: (<https://www.coursera.org/>)

- **Python for Everybody**  
<https://www.coursera.org/specializations/python>
- **Using Python to Access Web Data**  
<https://www.coursera.org/learn/python-network-data>
- **Introduction to Portfolio Construction and Analysis with Python**  
<https://www.coursera.org/learn/introduction-portfolio-construction-python>
- **Investment Management with Python and Machine Learning Specialization**  
<https://www.coursera.org/specializations/investment-management-python-machine-learning>
- **Fundamentals of Machine Learning in Finance**  
<https://www.coursera.org/learn/fundamentals-machine-learning-in-finance>
- **Innovations in Investment Technology: Artificial Intelligence**  
<https://www.coursera.org/learn/invest-tech>

## Courses in edX: (<https://www.edx.org/>)

- **Introduction to Programming Using Python**  
<https://www.edx.org/course/introduction-to-programming-using-python>
- **Introduction to Python: Absolute Beginner**  
<https://www.edx.org/course/introduction-to-python-absolute-beginner-1>
- **Introduction to Python: Fundamentals**  
<https://www.edx.org/course/introduction-to-python-fundamentals-1>
- **Python Data Structures**  
<https://www.edx.org/course/python-data-structures>

## Courses in Udemy: (<https://www.udemy.com/>)

- **Python for Finance: Investment Fundamentals & Data Analytics**  
<https://www.udemy.com/python-for-finance-investment-fundamentals-data-analytics/>
- **Python for Business and Finance**  
<https://www.udemy.com/course/complete-python-for-business-and-finance-bootcamp/>
- **Python for Financial Analysis and Algorithmic Trading**  
<https://www.udemy.com/course/python-for-finance-and-trading-algorithms/>

## Courses in Codecademy: (<https://www.codecademy.com/>)

- **Learn Python 3**  
<https://www.codecademy.com/learn/learn-python-3>
- **Analyze financial data with Python**  
<https://www.codecademy.com/learn/paths/finance-python>

## Courses in DataCamp: (<https://www.datacamp.com/>)

- **Introduction to Python**  
<https://www.datacamp.com/courses/intro-to-python-for-data-science>
- **Importing & Managing Financial Data in Python**  
<https://www.datacamp.com/courses/importing-managing-financial-data-in-python>
- **Manipulating Time Series Data in Python**  
<https://www.datacamp.com/courses/manipulating-time-series-data-in-python>
- **Introduction to Portfolio Analysis in Python**  
[https://www.datacamp.com/courses/introduction-to-portfolio-analysis-in-python?utm\\_medium=email&utm\\_source=customerio&utm\\_campaign=course\\_18408](https://www.datacamp.com/courses/introduction-to-portfolio-analysis-in-python?utm_medium=email&utm_source=customerio&utm_campaign=course_18408)
- **Machine Learning for Finance in Python**  
<https://www.datacamp.com/courses/machine-learning-for-finance-in-python>

## Are there any books on Python?

Yes, there are many, and more are being published. See the python.org wiki at: <https://wiki.python.org/moin/PythonBooks> for a list.

### Books:

- Brooker, Phillip, (2019), Programming with Python for Social Scientists, SAGE Publications.
- Dixon, Matthew F., Igor Halperin, Paul Bilokon, (2020), Machine Learning in Finance: From Theory to Practice, 1<sup>st</sup> Edition, Springer.
- Hilpisch, Yves, (2015), Python for Finance, O'Reilly.
- Hull, John C., (2021), Machine Learning in Business: An Introduction to the World of Data Science, 3<sup>rd</sup> Edition, Independently published.
- Lopez de Prado, Marcos, (2018), Advances in Financial Machine Learning, Wiley.
- Lopez de Prado, Marcos, (2020), Machine Learning for Asset Managers, Cambridge University Press.
- Nagel, Stefan, (2021), Machine Learning in Asset Pricing, Princeton University Press.
- Weiming, James Ma, (2019), Mastering Python for Finance: Implement advanced state-of-the-art financial statistical application using Python, 2<sup>nd</sup> Edition, Packt Publishing.
- Yuxing, Yan, (2017), Python for Finance: Apply Powerful Finance Models and Quantitative Analysis, 2<sup>nd</sup> Edition, Packt Publishing.

## **Example 1: (01 FinData\_from\_Yahoo)**

```
#First, make sure to have available all necessary libraries:
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import yfinance as yf
style.use('ggplot')

#Second, retrieve the data from, say, Yahoo
start = dt.datetime(2016,1,2)
end = dt.datetime(2021,10,31)

df = yf.download("AAPL", start, end)

print(df.head())
print(df.head(10))
print(df.tail())
print(df.tail(10))

#Save that data in file
df.to_csv('AAPL.csv')
```

## **Example 2: (02 ReadData\_from\_csvFile\_PlotData)**

```
#First, make sure to have available all necessary libraries:
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import yfinance as yf
style.use('ggplot')

#Read data from CSV file:
df = pd.read_csv('aapl.csv', parse_dates = True, index_col=0)
print(df.head())

#Plot data:
df.plot()
plt.show()

df['Adj Close'].plot()
plt.show()

print(df[['Open', 'High']].head())
```

### Example 3: (03 FinData\_Firms\_and\_Indeces)

```
#First, make sure to have available all necessary modules:
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import yfinance as yf
import numpy as np
style.use('ggplot')

#Second, retrieve the data from, say, Yahoo
start = dt.datetime(2012,12,31)
end = dt.datetime(2021,11,7)

#The list contains the 5 symbols for the stocks;
#Google (GOOGL), Facebook (FB), Microsoft (MSFT), Apple (AAPL), IBM (IBM), and
#the symbols for the Dow Jones Industrial Average and the S&P500

symbols = ["GOOGL", "FB", "MSFT", "AAPL", "IBM", "^DJI", "^GSPC"]

#Only the closing values of each data set:
data = pd.DataFrame()
for sym in symbols:
    data[sym] = yf.download(sym, start, end)['Close']
data = data.dropna()

print(data.head())
print(data.tail())

#plot data
data[['GOOGL','FB','MSFT','AAPL','IBM']].plot()
plt.show()

data[['FB','MSFT','AAPL','IBM']].plot()
plt.show()

#Save that data in file:
data.to_csv('DJI.csv')
```

### Example 4: (04 Rolling Std (Volatility))

```
#First, make sure to have available all necessary modules:
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import style
import pandas as pd
import yfinance as yf
import numpy as np
style.use('ggplot')

#Second, retrieve the data from, say, Yahoo:
start = dt.datetime(2002,12,31)
end = dt.datetime(2021,10,31)

df = yf.download("GOOGL", start, end)
```

```
print(df.head())
print(df.tail())

#Calculate the log returns:
df['Log_Return'] = np.log(df['Adj Close']/df['Adj Close'].shift(1))
print (df.head()['Log_Return'])
print(df.tail()['Log_Return'])

#Calculate the rolling standard deviation of the log returns (volatility):
df['Volatility'] = df['Log_Return'].rolling(window=252).std()*np.sqrt(252)

print(df.head())
print(df.tail())

#Save that data in file:
df.to_csv('GOOGL.csv')

df[['Adj Close','Volatility']].plot(subplots=True, color='blue', figsize=(8,6))
plt.show()
```