<https://towardsdatascience.com/how-to-build-a-simple-machine-learning-web-app-in-python-68a45a0e0291>

How to Build a Data Science Web App in Python

Part 1: Your First Web App in Just a Few Lines of Code

[Chanin Nantasenamat](https://towardsdatascience.com/@chanin.nantasenamat?source=post_page-----61d1bed65020----------------------)

[Jun 9](https://towardsdatascience.com/how-to-build-a-data-science-web-app-in-python-61d1bed65020?source=post_page-----61d1bed65020----------------------) · 4 min read

In this article, I will show you how you can quickly build a simple data-driven web app using the *streamlit* Python library in just a few lines of code.

As a Data Scientist or Machine Learning Engineer, it is important to be able to deploy our data science project. Traditional deployment of machine learning models with established framework such as Django or Flask may be a daunting and/or time-consuming task.

This article is based on a [YouTube video](https://www.youtube.com/watch?v=ZZ4B0QUHuNc) that I made on the same topic ([How to Build Your First Data Science Web App in Python](https://www.youtube.com/watch?v=ZZ4B0QUHuNc)) in which you can watch it alongside reading this article.

**Overview of the stock web app that we are building**

Today, we will be building a simple web app that displays the stock price and volume. This will require the use of two Python libraries namely streamlit and yfinance. Conceptually, the app will retrieve historical market data from Yahoo! Finance from the yfinance library. This data is saved into a dataframe and streamlit will use this data as input argument for displaying it as a line chart.

**Install prerequisite libraries**

In this tutorial, we will be using two Python libraries that require installation. These include streamlit and yfinance. You can easily do this via thepip installcommand as follows to install streamlit:

pip install streamlit

And do the same also for yfinance as follows:

pip install yfinance

# Code of the web app

Let’s take a look at the code of the web app that we are building today. You will see that there is less less than 20 lines of code (*i.e.* if not counting the comments then that trims it down to 14 lines of code where 3 of those lines are empty lines for aesthetic purpose).

import yfinance as yf

import streamlit as st

st.write("""

# Simple Stock Price App

Shown are the stock closing price and volume of Google!

""")

# https://towardsdatascience.com/how-to-get-stock-data-using-python-c0de1df17e75

#define the ticker symbol

tickerSymbol = 'GOOGL'

#get data on this ticker

tickerData = yf.Ticker(tickerSymbol)

#get the historical prices for this ticker

tickerDf = tickerData.history(period='1d', start='2010-5-31', end='2020-5-31')

# Open High Low Close Volume Dividends Stock Splits

st.line\_chart(tickerDf.Close)

st.line\_chart(tickerDf.Volume)

# Line-by-line explanation of the code

Let’s take some time to make some understanding of the above code.

* ***Lines 1 and 2***Imports yfinance and assigning it the alias of yf as well as importing streamlit and assigning it the alias of st.
* ***Lines 4–7***  
  Uses the st.write() function to print out text. These printed out text are written in markdown format.
* ***Lines 9–16***  
  Uses the yfinance library to retrieve historical market data from Yahoo! Finance.  
  \* *Line 11 —*Defines the ticker symbol as GOOGL.  
  \* *Line 13* — Creates the tickerDatavariable using the yf.Ticker()function that as the name imply allows access to the ticker data. It should be noted that the tickerData is a *Ticker object* and if we run tickerData as a command we get the following output yfinance.Ticker object <GOOGL>.  
  \* *Line 15* — Creates the tickerDfdataframe and defines the date range (from May 31, 2010 to May 31, 2020) and time period (1 day).
* ***Lines 18 and 19***  
  Uses the st.line\_chart() function to plot a line chart (using the close price from the *Close* and *Volume* columns from the tickerDf dataframe as defined on line 15.

# Running the web app

After saving the code into a file called myapp.py, fire up the command prompt (or Power Shell in Microsoft Windows) and run the following command (OR ANACONDA PROMPT!!!)

streamlit run myapp.py

Next, we should see the following message:

> streamlit run myapp.pyYou can now view your Streamlit app in your browser.Local URL: [http://localhost:8501](http://localhost:8501/)  
Network URL: <http://10.0.0.11:8501>

In a short moment, an internet browser window should pop-up and directs you to the created web app by taking you to

[http://localhost:8501.]<http://localhost:8501>

as shown below.

Congratulations! You have created your first web app in Python!

Customizing the web app

import yfinance as yf

import streamlit as st

st.write("""

# Simple Stock Price App

Shown are the stock \*\*closing price\*\* and \*\*\*volume\*\*\* of Google!

""")

# https://towardsdatascience.com/how-to-get-stock-data-using-python-c0de1df17e75

#define the ticker symbol

tickerSymbol = 'GOOGL'

#get data on this ticker

tickerData = yf.Ticker(tickerSymbol)

#get the historical prices for this ticker

tickerDf = tickerData.history(period='1d', start='2010-5-31', end='2020-5-31')

# Open High Low Close Volume Dividends Stock Splits

st.write("""

## Closing Price

""")

st.line\_chart(tickerDf.Close)

st.write("""

## Volume

""")

st.line\_chart(tickerDf.Volume)

Let’s take some time to make some understanding of the above code.

* ***Line 6***Notice that we bold the *“closing price”*using two asterisks in front and behind the phrase as follows: \*\*closing price\*\*. Also notice that we made the word “volume” to be both bold and in italic form by using three asterisks in front and behind the word as follows: \*\*\*volume\*\*\*.
* ***Line 18–20 and 22–25***Here we added a header titles in markdown format right before the *closing price*and the *volume* plots.

# Voila!

And now we have an updated web app that should automatically update right before your eyes

# About Me

I work full-time as an Associate Professor of Bioinformatics and Head of Data Mining and Biomedical Informatics at a Research University in Thailand. In my after work hours, I’m a YouTuber (AKA the [Data Professor](http://bit.ly/dataprofessor/)) making online videos about data science. In all tutorial videos that I make, I also share Jupyter notebooks on GitHub ([Data Professor GitHub page](https://github.com/dataprofessor/)).

## Connect with Me on Social Network

✅ YouTube: <http://youtube.com/dataprofessor/>  
✅ Website: [http://dataprofessor.org/](https://www.youtube.com/redirect?redir_token=w4MajL6v6Oi_kOAZNbMprRRJrvJ8MTU5MjI5NjQzN0AxNTkyMjEwMDM3&q=http%3A%2F%2Fdataprofessor.org%2F&event=video_description&v=ZZ4B0QUHuNc)(Under construction)  
✅ LinkedIn: <https://www.linkedin.com/company/dataprofessor/>  
✅ Twitter: <https://twitter.com/thedataprof>  
✅ FaceBook: [http://facebook.com/d](https://www.youtube.com/redirect?redir_token=w4MajL6v6Oi_kOAZNbMprRRJrvJ8MTU5MjI5NjQzN0AxNTkyMjEwMDM3&q=http%3A%2F%2Ffacebook.com%2Fdataprofessor%2F&event=video_description&v=ZZ4B0QUHuNc)  
✅ GitHub: [https://github.com/datap](https://github.com/dataprofessor/)