Q1. What is the distinction between a numpy array and a pandas data frame? Is there a way to convert between the two if there is?

The major differences between pandas DataFrame and numpy Array are listed below:

* Numpy arrays can be multi-dimensional whereas DataFrame can only be two-dimensional.
* Arrays contain similar types of objects or elements whereas DataFrame can have objects or multiple or similar data types.
* Both array and DataFrames are mutable.
* Elements in an array can be accessed using only integer positions whereas elements in a DataFrame can be accessed using both integer and index positions.
* DataFrames are mostly in the form of SQL tables and are associated with tabular data whereas arrays are associated with numerical data and computation.
* DataFrames can deal with dynamic data and mixed data types whereas arrays do not have the flexibility to handle such data.

Steps to Convert a NumPy Array to Pandas DataFrame:

Step 1: Create a NumPy Array

For example, let’s create the following NumPy array that contains only numeric data (i.e., integers):

import numpy as np

my\_array = np.array([[11,22,33],[44,55,66]])

print(my\_array)

print(type(my\_array))

Step 2: Convert the NumPy Array to Pandas DataFrame

You can now convert the NumPy array to Pandas DataFrame using the following syntax:

import numpy as np

import pandas as pd

my\_array = np.array([[11,22,33],[44,55,66]])

df = pd.DataFrame(my\_array, columns = ['Column\_A','Column\_B','Column\_C'])

print(df)

print(type(df))

Step 3 (optional): Add an Index to the DataFrame

What if you’d like to add an index to the DataFrame?

For instance, let’s add the following index to the DataFrame:

index = ['Item\_1', 'Item\_2']

So here is the complete code to convert the array to a DataFrame with an index:

import numpy as np

import pandas as pd

my\_array = np.array([[11,22,33],[44,55,66]])

df = pd.DataFrame(my\_array, columns = ['Column\_A','Column\_B','Column\_C'], index = ['Item\_1', 'Item\_2'])

print(df)

print(type(df))

Q2. What can go wrong when an user enters in a stock-ticker symbol, and how do you handle it?

A stock ticker is a report of the price of certain securities, updated continuously throughout the trading session by the various stock market exchanges. A "tick" is any change in the price of the security, whether that movement is up or down.

A ticker symbol, sometimes called a stock symbol, is an organization of letters or characters (or a combination of both) that represent particular securities that are either registered in a stock exchange or a public market. People who are interested in investing or trading securities use these symbols to place orders.

Q3. Identify some of the plotting techniques that are used to produce a stock-market chart.

There are many types of charts that are used for technical analysis. However, the four types that are most common are—line chart, bar chart, point and figure chart and candlestick chart.

Q4. Why is it essential to print a legend on a stock market chart?

he legend displays information about the points that are currently hovered over or, if none are hovered over, about the last points shown on the plot. The text of a legend item includes the name of a series and, depending on the series type, the value or values of the current or last point

Q5. What is the best way to limit the length of a pandas data frame to less than a year?

DataFrame. drop() method you can drop/remove/delete rows from DataFrame. axis param is used to specify what axis you would like to remove. By default axis = 0 meaning to remove rows.

Q6. What is the definition of a 180-day moving average?

A Simple Moving Average is adding up closing prices for a certain time period and then dividing the total by the number of days. The time period used is different and varies from trader to trader depending on their short-term or long-term investment strategy.

Q7. Did the chapter's final example use "indirect" importing? If so, how exactly do you do it?

A relative import specifies the resource to be imported relative to the current location—that is, the location where the import statement is. There are two types of relative imports: implicit and explicit. Implicit relative imports have been deprecated in Python 3, so I won’t be covering them here.

from .some\_module import some\_class

from ..some\_package import some\_function

from . import some\_class