

1.) Explain three-dimensional data indexing.

- NumPy arrays are high-performance data structures, better suited for mathematical operations than Python's native list data type. A three-dimensional (3D) array is composed of 3 nested levels of arrays, one for each dimension.

• We USE numpy.array() TO CREATE A 3D NUMPY ARRAY WITH SPECIFIC VALUES

- Call numpy.array(object) with object as a list containing x nested lists, y nested lists inside each of the x nested lists, and z values inside each of the y nested lists to create a x-by-y-by-z 3D NumPy array.

```
• a_3d_array = np.array ([[ [1, 2], [3, 4]], [[5, 6], [7, 8]]])
```

```
print(a_3d_array)
```

OUTPUT:- [[[1 2] [3 4]] [[5 6] [7 8]]]

2.) What's the difference between a series and a dataframe?

- Series can only contain single list with index, whereas dataframe can be made of more than one series or we can say that a dataframe is a collection of series that can be used to analyse the data

3.) What role does pandas play in data cleaning?

- Data cleaning with Pandas It is an essential skill of Data Scientists to be able to work with messy data, missing values, inconsistent, noise, or nonsensical data. To work smoothly python provides a built-in module Pandas.

4.) How do you use pandas to make a data frame out of n-dimensional arrays?

- import pandas as pd # Create the dataframe df = pd.DataFrame(numpy_array) df = pd.DataFrame(numpy_array, columns=['digits', 'words']) ... df = pd.DataFrame(numpy_array, index=['day1', 'day2', 'day3', 'day4'], columns=['digits', 'words'])

5.) Explain the notion of pandas plotting.

- Matplotlib is a Python plotting package that makes it simple to create two-dimensional plots from data stored in a variety of data structures including lists, numpy arrays, and pandas dataframes. Matplotlib uses an object oriented approach to plotting

- Ex:- import pandas as pd

```
import matplotlib.pyplot as plt  
df = pd.read_csv('data.csv')  
df.plot()  
plt.show()
```