1. Reshape the series ser into a dataframe with 7 rows and 5 columns

ser = pd.Series(np.random.randint(1, 10, 35))

1. Find the positions of numbers that are multiples of 3 from ser.

ser = pd.Series(np.random.randint(1, 10, 7))

1. From ser, extract the items at positions in list pos.

ser = pd.Series(list('abcdefghijklmnopqrstuvwxyz'))

pos = [0, 4, 8, 14, 20]

1. Stack ser1 and ser2 vertically and horizontally (to form a dataframe).

ser1 = pd.Series(range(5))

ser2 = pd.Series(list('abcde'))

1. Get the positions of items of ser2 in ser1 as a list.

ser1 = pd.Series([10, 9, 6, 5, 3, 1, 12, 8, 13])

ser2 = pd.Series([1, 3, 10, 13])

1. Compute the mean squared error of truth and pred series.

truth = pd.Series(range(10))

pred = pd.Series(range(10)) + np.random.random(10)

1. Change the first character of each word to upper case in each word of ser.

ser = pd.Series(['how', 'to', 'write', 'code?'])

1. Calculate the number of characters in each word in the following series

ser = pd.Series(['how', 'to', 'write', 'code?'])

1. Compute difference of differences between consequtive numbers of the following series?

ser = pd.Series([1, 3, 6, 10, 15, 21, 27, 35])

1. Convert the following series of date-strings to a timeseries?

ser = pd.Series(['01 Jan 2010', '02-02-2011', '20120303', '2013/04/04', '2014-05-05', '2015-06-06T12:20'])

1. Get the day of month, week number, day of year and day of week from ser.

ser = pd.Series(['01 Jan 2010', '02-02-2011', '20120303', '2013/04/04', '2014-05-05', '2015-06-06T12:20'])

1. Change ser to dates that start with 4th of the respective months.

ser = pd.Series(['Jan 2010', 'Feb 2011', 'Mar 2012'])

1. From ser, extract words that contain atleast 2 vowels.

ser = pd.Series(['Apple', 'Orange', 'Plan', 'Python', 'Money'])

1. Extract the valid emails from the series emails. The regex pattern for valid emails is provided as reference.

emails = pd.Series(['buying books at amazom.com', 'rameses@egypt.com', 'matt@t.co', 'narendra@modi.com'])

pattern ='[A-Za-z0-9.\_%+-][+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,4}](mailto:+@[A-Za-z0-9.-]+\\.%5bA-Za-z%5d%7b2,4%7d)'

1. Compute the mean of weights of each fruit.

fruit = pd.Series(np.random.choice(['apple', 'banana', 'carrot'], 10))

weights = pd.Series(np.linspace(1, 10, 10))

print(weight.tolist())

print(fruit.tolist())

#> [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0]

#> ['banana', 'carrot', 'apple', 'carrot', 'carrot', 'apple', 'banana', 'carrot', 'apple', 'carrot']