

4DAILY ONLINE ACTIVITIES SUMMARY

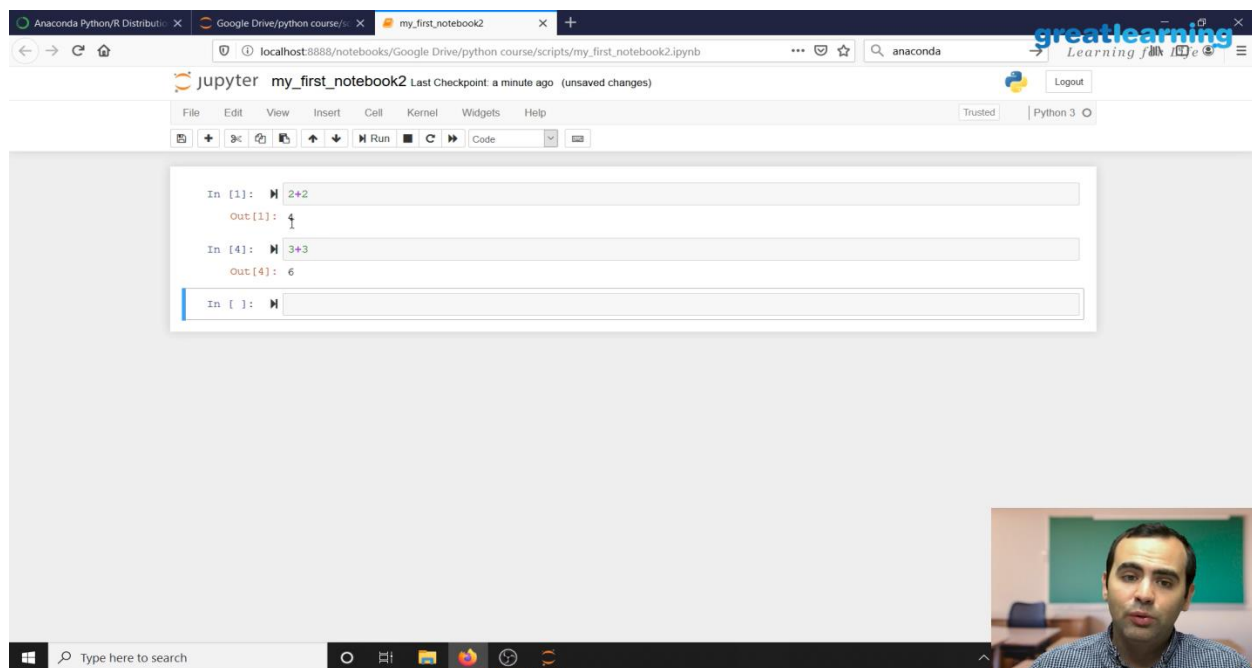
| | | | |
|---|---|---|------------------------|
| Date: | 11/06/2020 | Name: | M MAHAMMAD ASIF |
| Sem & Sec | 4th Sem & 'A' Sec | USN: | 4AL18CS045 |
| Online Test Summary | | | |
| Subject | - | | |
| Max. Marks | - | Score | - |
| Certification Course Summary | | | |
| Course | Python For Machine Learning | | |
| Certificate Provider | Great Learning | Duration | 5 Hours |
| Coding Challenges | | | |
| Problem Statement: 1. Java Program to Segregate Even and Odd numbers | | | |
| Status: Completed | | | |
| Uploaded the report in Github | | Yes | |
| If yes Repository name | | https://github.com/alvas-education-foundation/M_MAHAMMAD_ASIF | |
| Uploaded the report in slack | | Yes | |

Online Test Details: Today test was not conducted.

Certification Course Details: Today I started a new course that is python for machine learning which was about 5 hours of Duration . In that I had completed 3 hour of concepts.

In additional to this some other online courses I had completed, as a proof, I uploaded the Certificates in my other repository named “Completed course certificates.”

Snapshot:



olympus.greatlearning.in/courses/10899/pages/pandas-introduction-2?module_item_id=565941

greatlearning Learning for Life

Home Live Sessions Certificates

My Courses

Courses / Python for Machine Learning / Pandas - Introduction-4

Content

- Python for Machine Learning - Overview
- Course Overview
- Introduction to Python
- Practice Assessment
- Assessments
- Claim your course certificate

Pandas - Introduction-4

greatlearning Learning for Life

jupyter pandas_example Last Checked: Last Thursday at 1:26 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Python 3

Pandas

Pandas is a package used for managing data.

Pandas main use is that it creates 2 new data types for storing data: series and dataframe.

Think of a pandas dataframe like an excel spreadsheet that is storing some data. One column can have customer name, one column can have product sold name, another column can have price or quantity... Then the rows could be individual sales.

A dataframe is made up of several series. Each column of a dataframe is a series.

We can name each column and row of a dataframe.

A pandas dataframe is very similar to a data frame in R.

Similar to numpy arrays, a dataframe is a more robust data type for storing data than lists of lists. Dataframes are more flexible than numpy arrays.

A numpy array can create a matrix with all entries of the same data type. In a dataframe each column can have its own datatype.

That's not to say numpy arrays aren't useful. It is often easiest to convert some subset of a dataframe to a numpy array and then use that to do some math.

Pandas also has SQL-like functions for merging, joining, and sorting dataframes.

```
In [ ]: import pandas as pd
import numpy as np # numpy is not necessary for pandas, but we will use some np code in this example
# in general it's good practice to import all packages at the beginning

In [ ]: # first let's look at series - think of this as a single column of a spreadsheet
# each entry in a series corresponds to an individual row in the spreadsheet
# we use square brackets to access a single row in the series
```

4:20

1x

olympus.greatlearning.in/courses/10899/pages/lambda-functions?module_item_id=444828

greatlearning Learning for Life

Home Live Sessions Certificates

My Courses

Courses / Python for Machine Learning / Lambda Functions

Content

- Python for Machine Learning - Overview
- Course Overview
- Introduction to Python
- Practice Assessment
- Assessments
- Claim your course certificate

Lambda Functions

greatlearning Learning for Life

jupyter Python_functions_Class Last Checked: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Python 3

Use of user defined functions in machine learning

```
In [41]: # use to identify null values
import pandas as pd
import numpy as np

mpg_df = pd.read_csv("car-mpg.csv")
mpg_df = mpg_df.replace('?', np.nan) # pre-defined function replace
mpg_df['hp'] = mpg_df['hp'].astype(float64)
numeric_cols = mpg_df.drop('car_name', axis=1)
print(numeric_cols.head(10))

numeric_cols = numeric_cols.apply(lambda x: x.fillna(x.median()), axis=0) # lambda function
print(numeric_cols.head(10))
mpg_df['hp_median'] = numeric_cols['hp'].median()

mpg_df
```

| | mpg | cyl | displ | hp | wt | acc | yr | origin | car_type |
|---|------|-----|-------|-------|------|------|----|--------|----------|
| 0 | 18.0 | 8 | 387.0 | 130.0 | 3504 | 12.0 | 79 | 1 | 0 |
| 1 | 15.0 | 8 | 358.0 | 145.0 | 3693 | 11.5 | 78 | 1 | 0 |
| 2 | 18.0 | 8 | 318.0 | 150.0 | 3436 | 11.0 | 79 | 1 | 0 |
| 3 | 16.0 | 8 | 384.0 | 150.0 | 3423 | 12.0 | 79 | 1 | 0 |
| 4 | 17.0 | 8 | 302.0 | 140.0 | 3449 | 10.5 | 79 | 1 | 0 |
| 5 | 15.0 | 8 | 429.0 | 198.0 | 4341 | 10.0 | 79 | 1 | 0 |
| 6 | 14.0 | 8 | 454.0 | 220.0 | 4354 | 9.0 | 79 | 1 | 0 |
| 7 | 14.0 | 8 | 440.0 | 215.0 | 4312 | 8.5 | 79 | 1 | 0 |
| 8 | 14.0 | 8 | 455.0 | 225.0 | 4425 | 10.0 | 79 | 1 | 0 |

16:55

0:30

1x

Above are some Snapshots of today's certification course.

Coding Challenges Details: Today one Java problem was given by Prof Shilpa.I had solved that and uploaded the code in GitHub. The problem statements was:

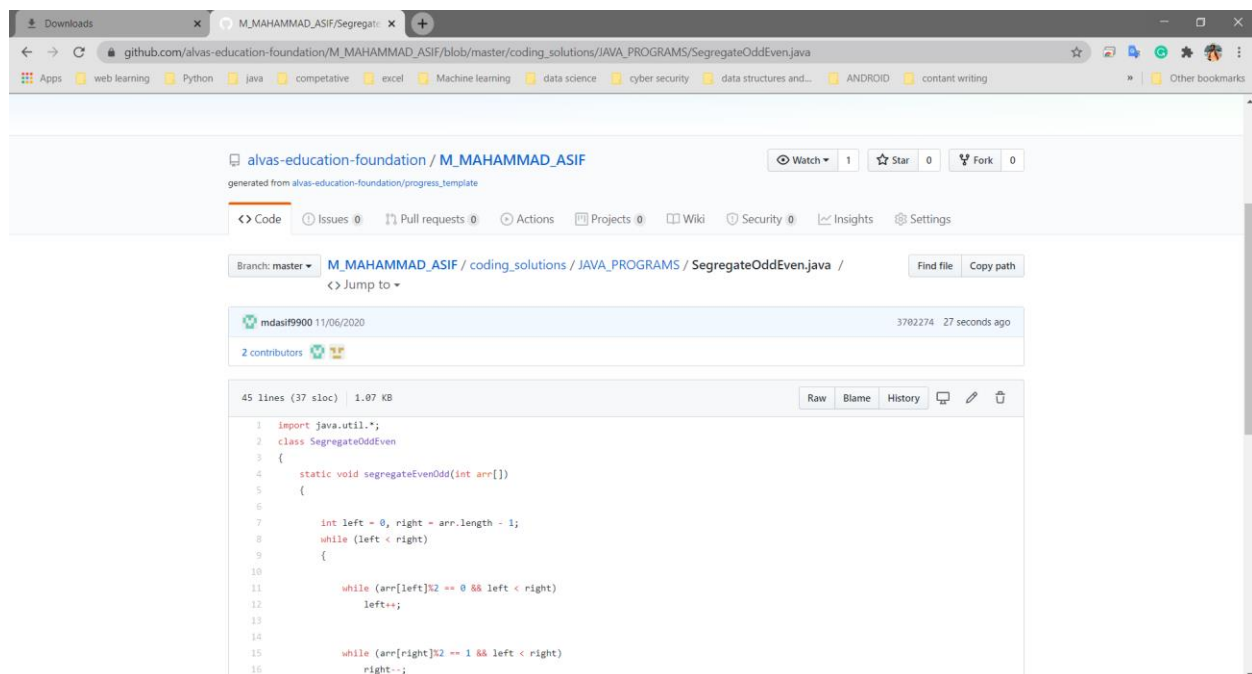
1. Given an array A[], write a function that segregates even and odd numbers. The functions should put all even numbers first, and then odd numbers.

Example:

Input = {12, 34, 45, 9, 8, 90, 3}

Output = {12, 34, 8, 90, 45, 9, 3}

Snapshot:



The screenshot shows a web browser displaying a GitHub repository page for 'alvas-education-foundation / M_MAHAMMAD_ASIF'. The repository is generated from a progress template. The file 'SegregateOddEven.java' is selected, showing its code. The code is a Java program that segregates even and odd numbers in an array using a two-pointer approach.

```
1 import java.util.*;
2 class SegregateOddEven
3 {
4     static void segregateEvenOdd(int arr[])
5     {
6
7         int left = 0, right = arr.length - 1;
8         while (left < right)
9         {
10
11             while (arr[left]%2 == 0 && left < right)
12                 left++;
13
14             while (arr[right]%2 == 1 && left < right)
15                 right--;
16
```

