

[Open in Colab](https://colab.research.google.com/github/prakash27/CSL522_ML/blob/master/assignments/Assignment_1_Python_Introduction.ipynb)

[(https://colab.research.google.com/github/prakash27/CSL522\_ML/blob/master/assignments/Assignment\_1\_Pytho](https://colab.research.google.com/github/prakash27/CSL522_ML/blob/master/assignments/Assignment_1_Python_Introduction.ipynb)

# CSB352: Data Mining

**Instructor** : [Dr. Chandra Prakash]

For more information visit the [class website (https://cprakash86.wordpress.com/csb352\_s21/)](https://cprakash86.wordpress.com/csb352_s21/).

# LAB\_1\_Assignment : Introduction to Python

## Due Date : 09-Jan-2021 Student Name: Rohit Byas

**Assignment Instructions**

You must save your as Assignment\_NO\_Yourname

**Assignment 1 :** To complete assignment one, first you have to add your name and roll no in the Google Colab Instructions section below and print it. Perform the 10 task given in the assignment and submit it over Microsoft Team.

Your source file will most likely end in **.pynb** if you are using a Jupyter notebook; however, it might also end in

**.py** if you are using a Python script.

# Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

In [1]:

**try**:

**from google.colab import** drive

%**tensorflow\_version** 2.x COLAB = **True**

print("Hello World")

print("Note: using Google CoLab")

**except**:

print("Hello NITD")

print("Note: not using Google CoLab")

COLAB = **False**

*# Print your name and Roll No.* print("Name: Rohit Byas sherwan") print("Roll No: 181210043")

*# Print the curent time*

**import datetime**

print(datetime.datetime.now())

Hello NITD

Note: not using Google CoLab Name: Rohit Byas sherwan

Roll No: 181210043

2021-01-06 01:15:25.151607

**Exercise 1.** Create following list:

data = [2,34,23,16,56,45,34,26,78,56,1,16]

Write a function to count number of elements in a list .

Now delete the duplicate values and print the list in ascending order.

In [2]:

*#Write your code here*

**def** count(arr):

**return** len(arr)

data = [2,34,23,16,56,45,34,26,78,56,1,16]

print(count(data))

data = list(set(data)) data.sort()

print(data)

12

[1, 2, 16, 23, 26, 34, 45, 56, 78]

**Exercise 2:** Take five numbers as input from the user and save into a list. Find the maximum of the list and sort the data in descending order.

In [3]:

*#Write your code here*

arr = list(map(int, input("Enter 5 no separated with spaces").split())) print("Max: ", max(arr))

arr.sort(reverse = **True**) print(arr)

Enter 5 no separated with spaces3 34 56 2 1

Max: 56

[56, 34, 3, 2, 1]

## Exercise 4:

1. Generate two arrays A1 and A2 of size 5 X 4 and 3 X 4 respectively using np.random()
2. Join them and make an array A3 of 8 X 4. Now append random numbers ranging between from 0 to 5 to make the fourth array A4 of size 10 X 10.
3. Print all the arrays and their transpose ( Transpose of 'A' can be obtained by 'A.T')

In [4]:

*#Write your code here*

**import numpy as np import random**

a1 = np.random.rand(5,4) a2 = np.random.rand(3,4) a3 = np.append(a1,a2)

a3 = np.reshape(a3, newshape = (8,4))

a4 = np.random.randint(0, high=5, size=(10,10), dtype='l')

print("Array 1: **\n**", a1)

print("Array 1 Transpose: **\n**",np.transpose(a1)) print("Array 2: **\n**", a2)

print("Array 2 Transpose: **\n**",np.transpose(a2)) print("Array 3: **\n**", a3)

print("Array 3 Transpose: **\n**",np.transpose(a3)) print("Array 4: **\n**", a4)

print("Array 4 Transpose: **\n**",np.transpose(a4))

Array 1:

[[0.77741873 0.82825236 0.04365063 0.72680121]

|  |  |  |  |
| --- | --- | --- | --- |
| [0.96019184 | 0.8520548 | 0.58432563 | 0.8911943 ] |
| [0.72082616 | 0.57365123 | 0.17358705 | 0.23070605] |
| [0.66811775 | 0.09790847 | 0.24771684 | 0.50289963] |
| [0.20197942 | 0.82159972 | 0.87824682 | 0.86563886]] |

Array 1 Transpose:

[[0.77741873 0.96019184 0.72082616 0.66811775 0.20197942]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [0.82825236 | 0.8520548 | 0.57365123 | 0.09790847 | 0.82159972] |
| [0.04365063 | 0.58432563 | 0.17358705 | 0.24771684 | 0.87824682] |
| [0.72680121 | 0.8911943 | 0.23070605 | 0.50289963 | 0.86563886]] |

Array 2:

[[0.16193668 0.71241554 0.37111388 0.91032102]

|  |  |  |
| --- | --- | --- |
| [0.08868736 0.13797175 | 0.38212671 | 0.11605065] |
| [0.1275112 0.81925618 | 0.07894434 | 0.252399 ]] |

Array 2 Transpose:

[[0.16193668 0.08868736 0.1275112 ]

|  |  |  |
| --- | --- | --- |
| [0.71241554 | 0.13797175 | 0.81925618] |
| [0.37111388 | 0.38212671 | 0.07894434] |
| [0.91032102 | 0.11605065 | 0.252399 ]] |

Array 3:

[[0.77741873 0.82825236 0.04365063 0.72680121]

|  |  |  |  |
| --- | --- | --- | --- |
| [0.96019184 | 0.8520548 | 0.58432563 | 0.8911943 ] |
| [0.72082616 | 0.57365123 | 0.17358705 | 0.23070605] |
| [0.66811775 | 0.09790847 | 0.24771684 | 0.50289963] |
| [0.20197942 | 0.82159972 | 0.87824682 | 0.86563886] |
| [0.16193668 | 0.71241554 | 0.37111388 | 0.91032102] |
| [0.08868736 | 0.13797175 | 0.38212671 | 0.11605065] |
| [0.1275112 | 0.81925618 | 0.07894434 | 0.252399 ]] |

Array 3 Transpose:

[[0.77741873 0.96019184 0.72082616 0.66811775 0.20197942 0.16193668

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0.08868736 | 0.1275112 ] |  | | | |
| [0.82825236 | 0.8520548 | 0.57365123 | 0.09790847 | 0.82159972 | 0.71241554 |
| 0.13797175 | 0.81925618] |  |  |  |  |
| [0.04365063 | 0.58432563 | 0.17358705 | 0.24771684 | 0.87824682 | 0.37111388 |
| 0.38212671 | 0.07894434] |  |  |  |  |
| [0.72680121 | 0.8911943 | 0.23070605 | 0.50289963 | 0.86563886 | 0.91032102 |
| 0.11605065 | 0.252399 ]] | | | | |
| Array 4: |  | | | | |
| [[2 4 3 3 3 | 0 0 1 3 1] | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [4 | 4 | 3 | 3 | 1 | 1 | 1 | 3 | 1 | 3] |
| [4 | 3 | 3 | 4 | 1 | 0 | 1 | 0 | 0 | 1] |
| [3 | 3 | 0 | 0 | 4 | 0 | 2 | 1 | 0 | 0] |
| [0 | 4 | 4 | 0 | 1 | 1 | 4 | 3 | 4 | 3] |
| [4 | 2 | 4 | 3 | 1 | 3 | 2 | 4 | 3 | 4] |
| [0 | 2 | 2 | 2 | 2 | 2 | 3 | 4 | 3 | 0] |
| [2 | 4 | 4 | 1 | 3 | 3 | 0 | 3 | 1 | 2] |
| [1 | 4 | 0 | 3 | 1 | 2 | 3 | 0 | 2 | 3] |
| [1 | 0 | 3 | 0 | 2 | 3 | 0 | 2 | 4 | 3]] |

Array 4 Transpose:

[[2 4 4 3 0 4 0 2 1 1]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [4 | 4 | 3 | 3 | 4 | 2 | 2 | 4 | 4 | 0] |
| [3 | 3 | 3 | 0 | 4 | 4 | 2 | 4 | 0 | 3] |
| [3 | 3 | 4 | 0 | 0 | 3 | 2 | 1 | 3 | 0] |
| [3 | 1 | 1 | 4 | 1 | 1 | 2 | 3 | 1 | 2] |
| [0 | 1 | 0 | 0 | 1 | 3 | 2 | 3 | 2 | 3] |
| [0 | 1 | 1 | 2 | 4 | 2 | 3 | 0 | 3 | 0] |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [1 | 3 | 0 | 1 | 3 | 4 | 4 | 3 | 0 | 2] |
| [3 | 1 | 0 | 0 | 4 | 3 | 3 | 1 | 2 | 4] |
| [1 | 3 | 1 | 0 | 3 | 4 | 0 | 2 | 3 | 3]] |

## Exercise 5: Create two dictionaries.

The first dictionary 'name' will contain first name(key) of a person and its hash value(value). The Second will contain hash value(key) and mobile no(value).

1. Add 5 entries.
2. Delete two entries by taking the input from user as the first name.

ii) Add two entries by taking the input as the first name and mobile no.

Hint: You can use remainder ( %) to obtain hash value.

In [10]:

*#Write your code here*

name = {

"Saksham": "saks0031", "Rohit": "rohi7451",

"Rahul": "rahu1817",

"Ayush":"ayus4538", "Vikas":"vika0636"

}

mobile = {

"saks0031": 8178340031,

"rohi7451": 7703987451,

"rahu1817": 9419201817,

"ayus4538": 9501084538,

"vika0636": 7792850636,

}

print(name) print(mobile)

**for** i **in** range (2):

name\_input = input("Enter the name to remove: ")

hash\_value = name[name\_input]

**del** name[name\_input]

**del** mobile[hash\_value]

print(name) print(mobile)

**for** i **in** range (2):

name\_input = input("Enter the name: ")

mobile\_input = int(input("Enter the mobile: "))

name[name\_input] = name\_input.lower()[:4] + str((mobile\_input%**10000**))

mobile[name\_input.lower()[:4] + str((mobile\_input%**10000**))] = mobile\_input

{'Saksham': 'saks0031', 'Rohit': 'rohi7451', 'Rahul': 'rahu1817', 'Ayush': 'a yus4538', 'Vikas': 'vika0636'}

{'saks0031': 8178340031, 'rohi7451': 7703987451, 'rahu1817': 9419201817, 'ayu

s4538': 9501084538, 'vika0636': 7792850636}

Enter the name to remove: Saksham Enter the name to remove: Vikas

{'Rohit': 'rohi7451', 'Rahul': 'rahu1817', 'Ayush': 'ayus4538'}

{'rohi7451': 7703987451, 'rahu1817': 9419201817, 'ayus4538': 9501084538}

Enter the name: Saksham

Enter the mobile: 8178340031 Enter the name: Vikas

Enter the mobile: 7792850636

# PART 2: Introduction to NumPy

**Exercise 6:** Write a NumPy program to create an element-wise comparison (greater, greater\_equal, less and less\_equal) of two given arrays

In [11]:

*#Write your code here*

**import numpy as np**

a = np.random.randint(0, high=50, size=(10), dtype='l') b = np.random.randint(0, high=50, size=(10), dtype='l')

print("Original numbers:") print(a)

print(b)

print("Comparison - greater") print(np.greater(a, b))

print("Comparison - greater\_equal") print(np.greater\_equal(a, b))

print("Comparison - less") print(np.less(a, b))

print("Comparison - less\_equal") print(np.less\_equal(a, b))

Original numbers:

|  |  |  |  |
| --- | --- | --- | --- |
| [30 26 30 9 | 31 28 36 | 0 | 20 3] |
| [48 46 19 16 | 29 17 43 | 4 | 18 39] |
| Comparison - | greater |  |  |

[False False True False True True False False True False]

Comparison - greater\_equal

[False False True False True True False False True False]

Comparison - less

[ True True False True False False True True False True]

Comparison - less\_equal

[ True True False True False False True True False True]

**Exercise 7:** Write a NumPy program to create an array with the values 100, 71, 113, 1050 and determine the size of the memory occupied by the array.

In [17]:

*#Write your code here*

arr = np.array([100, 71, 113, 1050]) print(arr.nbytes)

16

**Exercise 8:** Write a NumPy program to get the powers ( x ^ 3 ) of an array values element-wise Expected Output: Original array [1 2 3 4 5 ] Output array: [ 1 8 27 64 125 ]

In [18]:

*#Write your code here*

arr = np.array([1, 2, 3, 4, 5])

print(np.power(arr, 3))

[ 1 8 27 64 125]

**Exercise 9:** Write a NumPy program to get the floor, ceiling and truncated values of the elements of a numpy array. Sample Output: Original array: [-1.3, -1.15, -0.1, 0.12, 1.7, 0.9, 1.1]

In [19]:

*#Write your code here*

**import numpy as np**

arr = np.array([-1.3, -1.15, -0.1, 0.12, 1.7, 0.9, 1.1])

print(arr)

print("Floor:")

print(np.floor(arr)) print("Ceil:")

print(np.ceil(arr)) print("Truncated:") print(np.trunc(arr))

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| [-1.3 -1.15 | -0.1 | 0.12 | 1.7 | 0.9 | 1.1 ] |
| Floor: |  |  |  |  |  |
| [-2. -2. -1. | 0. 1. 0. | | 1.] |  |  |
| Ceil: |  |  |  |  |  |
| [-1. -1. -0. | 1. 2. 1. | | 2.] |  |  |
| Truncated: |  |  |  |  |  |
| [-1. -1. -0. | 0. 1. 0. | | 1.] |  |  |

**Exercise 10:** Write a program in python to display prime numbers from x to y (here x and y are user given values).

In [23]:

*#Write your code here*

**def** checkprime(n):

**for** i **in** range (2, n):

**if** n % i == 0:

**return** 0

**return** 1

In [25]:

x = int(input("Enter x: "))

y = int(input("Enter y: "))

**for** i **in** range (x, y+1):

**if** checkprime(i): print(i)

Enter x: 12

Enter y: 35

13

17

19

23

29

31

# Observation \ Comments

In [26]:

*#Write your observation\ Learning here*

*"""*

*I have used Python in competitive progamming, web development and also done a deep learning project in python. So working with this notebook is good and I am exicted to start this course and learn new thing and go much deeper in programming field.*

*"""*

Out[26]: ‘ I have used Python in competitive progamming, web development and also done a deep learning project in python. So working with this notebook is good and I am exicted to start this course and learn new thing and go much deeper in programming field.’

**Hint :**

[Reference Book to learn python: https://anandology.com/python-practice-book/getting-started.html (https://anandology.com/python-practice-book/getting-started.html)](https://anandology.com/python-practice-book/getting-started.html)

In [ ]: